PRIMAL-GMM: PaRametrIc MAnifold Learning of Gaussian Mixture Models

Ziquan Liu, Lei Yu, Janet H. Hsiao, Antoni B. Chan,

Abstract—We propose a ParametRIc MAnifold Learning (PRIMAL) algorithm for Gaussian Mixtures Models (GMM), assuming that GMMs lie on or near to a manifold of probability distributions that is generated from a low-dimensional hierarchical latent space through parametric mappings. Inspired by Principal Component Analysis (PCA), the generative processes for priors, means and covariance matrices are modeled by their respective latent space and parametric mapping. Then, the dependencies between latent spaces are captured by a hierarchical latent space by a linear or kernelized mapping. The function parameters and hierarchical latent space are learned by minimizing the reconstruction error between ground-truth GMMs and manifold-generated GMMs, measured by Kullback-Leibler Divergence (KLD). Variational approximation is employed to handle the intractable KLD between GMMs and a variational EM algorithm is derived to optimize the objective function. Experiments on synthetic data, flow cytometry analysis, eye-fixation analysis and topic models show that PRIMAL learns a continuous and interpretable manifold of GMM distributions and achieves a minimum reconstruction error.

Index Terms—Dimensionality Reduction and Manifold Learning, Gaussian Mixture Models, Interpretability, Unsupervised Learning, Probabilistic Models

1 Introduction

Probabilistic models are effective representations for real-world data in the presence of uncertainties. For example, hidden Markov models (HMMs), a probabilistic model for series of observations whose hidden states are a Markov process, are widely used in speech recognition [1] and sequence analysis [2]; linear dynamical systems (dynamic textures, DTs) are used to describe a video since they can abstract complex patterns of motions and appearance [3]. Our main interest in this paper is Gaussian Mixture Models (GMMs), a class of probabilistic models that represents multimodal data, where each mode is a Gaussian density represented by its mean and covariance. Moreover, GMMs can approximate any continuous probability density function as probability theory shows [4, 5]. Hence, there are numerous applications of GMMs in all kinds of machine intelligence research fields. In computer vision, GMMs are considered as a *universal visual vocabulary* of image patches [6–9]; in natural language processing, researchers use GMMs to represent a mixture of topics [10–12] or grammar rules [13]; in speech recognition, the emission probability from a phoneme to a speech segment is often modeled as a GMM [1]. Previous works have proposed to cluster probabilistic models to obtain hierarchical representations of data, which can be further employed in retrieval, annotation, indexing and codebook generation [14–17].

While clustering probabilistic models (PMs) gives hierarchical representations, it cannot learn a *continuous* and *interpretable* manifold on which we can see the continuous change between PMs. In various application domains, such as medical diagnosis [18, 19], behavior analysis [20] and text analysis [11, 12], such an interpretable manifold provides better insights into the differences between subjects (the GMMs) and the underlying mechanisms. Specifically, suppose we collect data from N subjects and learn N GMMs as their representation. Clustering GMMs will give several common patterns among the subjects, from which we only obtain limited discrete representations. Alternatively, if the GMMs are embedded into a low-dimensional manifold, the coordinates of subjects on that manifold provide a continuous and low-dimensional latent space, and the relationship between subjects and their properties, e.g., healthy conditions in medical diagnosis and subject age/performance in behavior analysis, can be revealed via correlation analysis. In other words, if we reduce the dimensionality of GMMs so that GMMs can be represented in a low-dimensional latent space, the latent space will reflect how subjects' GMMs are correlated with their other properties. Furthermore, if we are able to reconstruct GMMs from the low-dimensional latent space, the continuous change along specific directions on the manifold can be readily obtained, providing interpretability of the hidden mechanisms of the revealed correlations. Note that here we are concerned with learning a low-dimensional manifold of GMM *distributions* (where the GMM itself is the data sample), as opposed to learning a manifold of vector data sampled from a GMM, as in [21].

Despite its importance, manifold learning for PMs is not well-explored. There are two general approaches for manifold learning of PMs: kernel embedding and latent variable models. Kernel embedding explicitly models the mapping from input PMs to latent variables using a kernel function (or distance function). Hence, PMs can be embedded into a low-dimensional space by a suitably-defined kernel function over probability distributions. While the forward mapping from input space to latent space is readily available, finding an inverse mapping in kernel embedding, known as the pre-image problem [22], is often difficult. Thus the interpretability of kernel embeddings is often limited. In contrast, latent variable models learn the generative process, i.e. generative mapping from low-dimensional latent variables to high-dimensional variables. But such generative

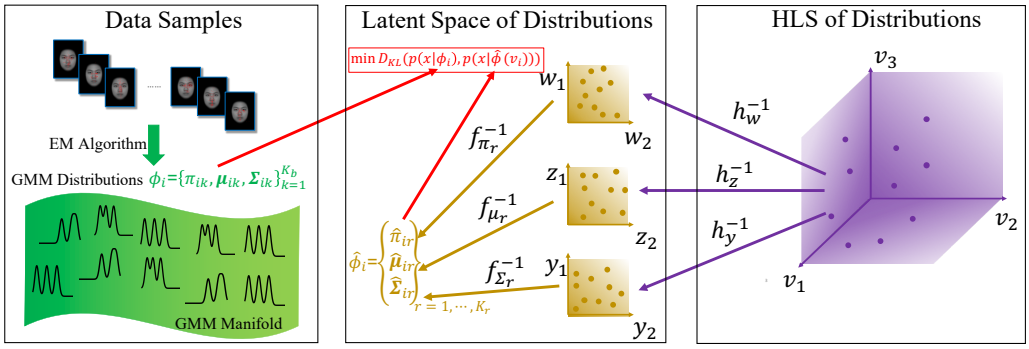


Fig. 1: Learning a Parametric Manifold for GMMs. (left) Given a set of GMM distributions $\{\phi_i\}_{i=1}^N$, parametrized by component priors, means, and covariances $\phi_i = \{\pi_{ik}, \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}\}_{k=1}^{K_b}$, the goal is to learn a low-dimensional manifold spanning the GMM distributions. (middle) On the manifold, the GMM parameters $\hat{\phi}_i = \{\hat{\pi}_{ir}, \hat{\boldsymbol{\mu}}_{ir}, \hat{\boldsymbol{\Sigma}}_{ir}\}_{k=1}^{K_r}$ are generated from a latent space $(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y})$ by their corresponding generative functions $(f_{\pi_r}^{-1}(\boldsymbol{w}_i), f_{\mu_r}^{-1}(\boldsymbol{z}_i), f_{\Sigma_r}^{-1}(\boldsymbol{y}_i))$. The parameters of the generative functions are obtained by minimizing the reconstruction loss (KL divergence) between the given GMMs ϕ_i and their reconstructions $\hat{\phi}_i$. (right) We further assume that the latent space $(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y})$ is generated from a hierarchical latent space (HLS) $\boldsymbol{v}$ by parametric functions $h^{-1}(\boldsymbol{v})$ to further reduce the dimension, which models dependencies among the component priors, means, and covariances.

models only take vectors as input and are not directly applicable to PMs. AutoEncoders (AEs) [23] are another type of nonlinear embedding method with latent variables. AEs first map input vectors to a latent space (encoder) and then reconstruct input vectors from latent variables (decoder). Although AEs are able to model complex forward and inverse mappings by neural networks explicitly, it is unclear how to handle inputs consisting of multi-modal distributions like GMMs, except vectorizing such input.

In this paper, we propose to learn a smooth and interpretable low-dimensional manifold for GMMs such that the generative mapping from the low-dimensional latent space and the statistical manifold can be obtained easily and probabilistic properties of GMMs are well respected. Inspired by PCA, we propose a *parametric* approach for learning a manifold of distributions. The GMM parameters for the component priors, means and covariances $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K_b}$ are each modeled by their own *principal axes*, i.e. mappings from latent variables to the high-dimensional manifold. We minimize the KL divergence between the original GMMs and their reconstructions from the latent space, which makes our method more accommodating to probabilistic models compared to latent variable models. However, it is well-known that the KL divergence between GMMs is intractable so we resort to variational approximation to obtain an upper-bound of the KL loss. We propose a variational EM optimization algorithm: alternatively optimizing variational parameters and manifold parameters. To avoid local minima, Metropolis-Hastings sampling [24, 25] is used to introduce stochasticity into the optimization process. Finally, a Hierarchical Latent Space (HLS) is constructed to capture dependencies between the latent spaces for the priors, means and covariances. Our contributions are 4-fold: 1) we propose a novel learning framework, PaRametrIc MAnifold Learning of GMMs (PRIMAL-GMM), to learn a smooth and interpretable low-dimensional manifold of GMMs; 2) we derive an EM-style optimization algorithm to learn such a manifold and use Monte Carlo sampling to avoid local minima ; 3) we propose to learn a kernelized hierarchical latent space to model nonlinear dependencies between different GMM parameter spaces; 4) we empirically show the efficacy of PRIMAL-GMM on various applications, including eye-fixation data analysis, flow cytometry analysis and topic model visualization.

A preliminary conference version of this work was published as [26]. Compared to [26], we have three main improvements in this paper: (1) we propose a kernelized HLS to model nonlinear dependencies in GMM parameters; (2) we propose a parameterization for diagonal-covariance GMMs and provide a corresponding implementation in Tensorflow [27] to enhance scalability and flexibility; (3) we test our method on several real-world datasets of different modality such as text documents.

The remainder of this paper is organized as follows. We discuss related work in Section 2, and propose the learning framework and optimization algorithm in Section 3. In Section 4, we extend the linear HLS to a kernel HLS. Finally, Section 5 presents experimental evaluations of PRIMAL-GMM and several baseline methods on one synthetic dataset and four real-world applications.

2 Related Work

Given a set of probabilistic models (PMs), with each PM representing one subject in the dataset, the relationship among the PMs can be uncovered through clustering, to obtain discrete groups of common models, or through dimensionality reduction (manifold learning), to obtain a latent space where directions in the latent space correspond to changes in PM.

2.1 Clustering for Probabilistic Models

The hierarchical EM (HEM) algorithm is the seminal work in clustering PMs [15], and was first proposed to cluster Gaussian distributions. The Gaussians are collected into a "base" GMM, from which a "reduced" GMM is estimated with a fewer number of components. The components in the reduced GMM serve as the representative Gaussians for the clusters, and the cluster memberships map between base and reduced Gaussian components. To avoid the high computational cost, virtual samples are generated from the base GMM and a closed-form solution is derived for the parameter updates of the reduced GMM, expressed in terms of the parameters of the base GMM. [28] proposed to minimize a distance between a base GMM and a reduced GMM that neglects component priors of the reduced GMM and obtained a hard-clustering algorithm. [29] proposed to minimize the KL divergence between the base GMM and the reduced GMM using a variational approximation, since the

KL divergence between two mixture models is intractable, also resulting in a hard clustering algorithm. Recently [14] derived a tighter upper-bound of the KL divergence and proposed a density-preserving HEM (DPHEM), i.e., the reduced GMM preserves the density of the base GMM. Besides GMMs, HEM was extended to cluster time-series PMs: Linear Dynamical Systems [16, 30] and HMMs [17, 31].

These clustering methods only give a discrete representation of the set of PMs, i.e., a finite set of representative models and cluster assignments. In contrast, we propose to learn a manifold of PMs, which is a continuum of representative models specified by the latent space coefficients. Hence, our model can better visualize changes in the structure of the PMs. In our learning algorithm, we adopt a variational approximation to the expected log-likelihood, which is similar to DPHEM [14]. The main difference is that DPHEM takes a single input GMM and reduces the number of components to obtain a single output GMM. In contrast, our formulation takes a *set* of GMMs and embeds them into a parametric manifold by minimizing the reconstruction loss of the GMMs. DPHEM is a special case of our framework when there is one input GMM and $K_m < K_b$, where K_m and K_b are the number of components in the reconstruction GMM and the input GMM. Also, the EM algorithm in our paper has no closed-form solution in the M-step, whereas the simpler M-step of DPHEM has a closed-form solution.

2.2 Manifold Embedding for Probabilistic Models

As we discussed in Section 1, kernel embeddings and latent variable models can be used for manifold learning of PMs. Kernel PCA [32] can be used with KL kernel [33] or probability product kernel [34] to perform nonlinear dimensionality reduction for a set of GMMs. Based on information geometry [35], Fisher Information Nonparametric Embedding (FINE) is proposed, which computes geodesics on the Riemannian manifold of probabilistic distributions, and then uses multi-dimensional scaling (MDS) [36] to obtain embeddings [18]. The advantage of these kernel methods is that the forward mapping from distributions to embedding coordinates can be obtained explicitly. However, the disadvantage is that the generative mapping from embedding coordinates to distributions is difficult and requires solving the pre-image problem, which hinders interpretation of the embedding space and its relationship with the input space. In contrast to kernel methods, our method explicitly constructs the generative mapping from latent space to probability space. Gaussian Process Latent Variable Models (GPLVM) [37] is a representative latent variable model, which assumes the non-linear generative mapping is a Gaussian process. However, the high-dimensional variables are treated as vectors, and thus GPLVM cannot naturally represent structured non-vector data, such as probability distributions. While it is possible to also kernelize the high-dimensional variable, this leads to the same pre-image problem as KPCA above. Similar to GPLVM, our method is also a generative model, but in contrast to GPLVM, we construct an explicit parametric mapping from the latent space to probability distributions. Another drawback of GPLVM is that it assumes no dependencies among different dimensions of input vectors, while PRIMAL models dependencies between input GMM parameters via a hierarchical latent space.

Mixed models are able to accommodate correlation between individual samples by introducing random effects into the prediction distributions. Thus, the distribution of random effects provides certain individual properties, depending on the corresponding covariates. Mixed hidden Markov models (MHMM) [38] is a generalization of mixed models to capture the correlation between multiple processes, where the random effects can be interpreted as the hidden random variables for individual processes, i.e., latent variables for HMMs. Mixed GMM (MGMM) can be obtained from MHMM by setting the number of state to be 1 and the emission probabilistic model to be a GMM. In this sense, MGMM learns mixed GMMs and corresponding latent variables simultaneously from samples of a collection of subjects. In contrast, PRIMAL learns latent variables from given GMMs, where the predictor model is either a linear or kernel function and the link function is f^{-1}'s mapping latent variables to GMM parameters. We give an ablation study in Section 4 to compare the efficacy of PRIMAL and its MGMM variant which learns a GMM manifold from observation data directly.

3 Parametric Manifold Learning of Gaussian Mixture Models

Let $\{\pi_{ik}, \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}\}_{k=1}^{K_b}$ be the parameters of the ith GMM with K_b components, where π_{ik} is the prior probability of the kth component, and $\boldsymbol{\mu}_{ik} \in \mathbb{R}^D$ and $\boldsymbol{\Sigma}_{ik} \in \mathbb{S}_+^{D\times D}$ are the mean and covariance matrix of the kth component. The probability distribution for a GMM is $p_i(\boldsymbol{x}) = \sum_{k=1}^{K_b} \pi_{ik}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})$, where $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-d/2}|\boldsymbol{\Sigma}|^{-1/2}\exp(\frac{-1}{2}\|\boldsymbol{x}-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}})$ is the multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and $\|\boldsymbol{x}-\boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}} = (\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$. Our goal is to learn a generative mapping from a low-dimensional latent space to the statistical manifold of GMMs, i.e., $f^{-1} : (\boldsymbol{w}_i, \boldsymbol{z}_i, \boldsymbol{y}_i) \to p_i(\boldsymbol{x})$ where $\boldsymbol{w}_i \in \mathbb{R}^{d_w}$, $\boldsymbol{z}_i \in \mathbb{R}^{d_z}$, $\boldsymbol{y}_i \in \mathbb{R}^{d_y}$ are the latent variables for component prior, mean and covariance respectively.

3.1 Parametric Manifold for GMMs

In contrast to kernel embeddings, here we focus on explicitly constructing the generative mapping from the latent space to the statistical manifold. Following PCA reconstruction, we define a set of "principal axes" and corresponding "coefficients" $(\boldsymbol{w}_i, \boldsymbol{z}_i, \boldsymbol{y}_i)$ for each GMM parameter (prior, mean, covariance), from which the parameters can be reconstructed. The "principal axes" and "coefficients" in our method is generalized to parameterized principal functions and latent variables. The latent space variables are used to reconstruct a GMM with K_m components (possibly different from K_b)[1], with parameters $\{\hat{\pi}_{ir}, \hat{\boldsymbol{\mu}}_{ir}, \hat{\boldsymbol{\Sigma}}_{ir}\}_{r=1}^{K_m}$. The rth component of the reconstructed GMM is defined by rth principal functions and latent variables, denoted as $f^{-1}_{\pi_r} : \boldsymbol{w}_i \to \hat{\pi}_{ir}$, $f^{-1}_{\mu_r} : \boldsymbol{z}_i \to \hat{\boldsymbol{\mu}}_{ir}$ and $f^{-1}_{\Sigma_r} : \boldsymbol{y}_i \to \hat{\boldsymbol{\Sigma}}_{ir}$. Note that latent variables $(\boldsymbol{w}_i, \boldsymbol{z}_i, \boldsymbol{y}_i)$ are shared among all reconstructed components while each reconstructed component r has its own principal functions $(f^{-1}_{\pi_r}, f^{-1}_{\mu_r}, f^{-1}_{\Sigma_r})$.

1. Although it is possible for K_m to be different from K_b in principle, we set $K_m = K_b$ in our experiment to better reconstruct the manifold, as in the synthetic experiment. We also suggest users to set $K_m = K_b$ for the same reason.

The next question is how to choose the parameterizations for the principal functions. Many existing works have proposed parameterizations for Gaussian parameters so that they satisfy constraints for those parameters naturally, i.e., priors must lie in a simplex and covariance matrices must be positive definite [7, 39, 40]. Here we employ different parameterizations for full-covariance GMMs and diagonal-covariance GMMs because the two kinds of GMMs often come up in problems of different scales: diagonal matrices are a common setting in large scale problems, while full covariance matrices are often used in small datasets.

3.1.1 Full-Covariance GMM Parameterization

If a dataset contains N subjects and we learn a GMM for each subject, the total number of parameters of full covariance matrices will be $O(D^2K_bN)$. Therefore, full covariance matrices are not practical for a large scale dataset with hundreds of dimensions due to the D^2 term. But for small datasets like eye-fixation and flow-cytometry in our experiment, full covariance matrices are necessary to represent the correlation between different dimensions. Note that the estimation of full covariance matrices requires a large number of samples for each subject. Here we propose a parameterization for full-covariance GMMs so that priors and covariance matrices fulfill their constraints automatically. The principal functions are

$$\begin{aligned}\hat{\pi}_{ir} &= f^{-1}_{\pi_r}(\boldsymbol{w}_i;\boldsymbol{a}_r) = \frac{\sigma(\boldsymbol{w}_i^T\boldsymbol{a}_r)}{\sum_{n=1}^{K_m}\sigma(\boldsymbol{w}_i^T\boldsymbol{a}_n)},\\ \hat{\boldsymbol{\mu}}_{ir} &= f^{-1}_{\mu_r}(\boldsymbol{z}_i;\boldsymbol{m}_r,\boldsymbol{b}_r) = (\sum_{l=1}^{d_z} z_{il}\boldsymbol{m}_{rl}) + \boldsymbol{b}_r, \qquad (1)\\ \hat{\boldsymbol{\Sigma}}^{-1}_{ir} &= f^{-1}_{\Sigma_r}(\boldsymbol{y}_i;\boldsymbol{C}_r,\beta_r) = \sum_{l=1}^{d_y}\log(1+\exp(y_{il}))\boldsymbol{C}_{rl}\boldsymbol{C}_{rl}^T + \beta_r^2\boldsymbol{I},\end{aligned}$$

where $\boldsymbol{a}_r \in \mathbb{R}^{d_w}$, $\boldsymbol{m}_{rl}, \boldsymbol{b}_r \in \mathbb{R}^D$, $\boldsymbol{C}_{rl} \in \mathbb{R}^{D\times D}$, $\beta_r \in \mathbb{R}$, $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function, y_{il}, z_{il} are the l-th coefficients of $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$.

Similar to PCA, the mean $\hat{\boldsymbol{\mu}}_{ir}$ is a linear combination of principal axes $\boldsymbol{m}_{rl}$, weighted by z_{il}, and an offset vector $\boldsymbol{b}_r$. The reconstructed precision matrix $\hat{\boldsymbol{\Sigma}}^{-1}_{ir}$ is a linear combination of $\boldsymbol{C}_{rl}\boldsymbol{C}_{rl}^T$ and an offset $\beta_r^2\boldsymbol{I}$. The reason for reconstructing the precision matrix in this way is threefold. First, the positive definite constraint of $\hat{\boldsymbol{\Sigma}}_{ir}$ is naturally fulfilled since the weights $\log(1+\exp(y_{il}))$ are always non-negative. Second, when the latent variable $\boldsymbol{y}_i \ll \boldsymbol{0}$, then the precision matrix will be a "default" value (i.e., a fixed level of uncertainty) of $\beta_r^2\boldsymbol{I}$. For increasing values of the latent variable $\boldsymbol{y}_i$, the precision will increase, i.e., the covariance (uncertainty) decreases. Thus the latent variable naturally interpolates between different shapes of covariance matrices and a default covariance. Third, the gradients are easier to compute when defining the reconstruction through the precision matrix. For priors $\hat{\pi}_r$, empirically we found that using the sigmoid function has more stable training, c.f., the softmax function. Using the linear parametrization, the softmax function saturates too quickly to 0 or 1 due to the exponential function, and makes it difficult to learn to predict component probabilities between 0 and 1. Also note that the probability constraints (non-negative and sum to 1) on the prior are naturally fulfilled by the formulation.

Note that there is no need to define an explicit correspondence between the rth component of the reconstructed GMM and kth component of the input GMM, since the learning algorithm uses the reconstruction loss between the whole input GMM and the whole reconstruction GMM – the ordering of the components in the input GMMs will not affect the embedding. Finally, PCA is a special case of our formulation in (1) with only one component $K_b = K_m = 1$, and the latent variable $\boldsymbol{y}_i \to -\infty$ and β_r is a constant (see Appendix A). From this perspective, our method is a generalization of vanilla PCA to probabilistic models.

3.1.2 Diagonal-Covariance GMM Parameterization

For large scale datasets such as images [7], audio segments [41] and topic models [12], diagonal covariance is a practical setting as a result of the poor scalability of full covariance matrices with respect to vector dimensions. Here we use a different parameterization for diagonal covariance matrices to exploit the diagonal property so that the computation is much faster than that of full covariance matrices parameterization. A two-layer neural network (NN) with an exponential function output is used as the parameterization of $f^{-1}_{\Sigma_r}(\boldsymbol{y}_i;\boldsymbol{C}_r^{(NN)})$, i.e.,

$$\begin{aligned}\hat{\boldsymbol{\Sigma}}^{-1}_{ir} &= f^{-1}_{\Sigma_r}(\boldsymbol{y}_i;\boldsymbol{C}_r^{(NN)}) \qquad (2)\\ &= \mathrm{Diag}(\exp(\boldsymbol{W}_r^{(\boldsymbol{C},2)}\cdot\mathrm{ReLU}(\boldsymbol{W}_r^{(\boldsymbol{C},1)}\boldsymbol{y}_i + \boldsymbol{h}_r^{(\boldsymbol{C},1)}) + \boldsymbol{h}_r^{(\boldsymbol{C},2)}))\end{aligned}$$

where Diag is the operator to form a diagonal matrix from a vector, ReLU is the rectified linear unit, and the NN parameters are $\boldsymbol{C}_r^{(NN)} = \{\boldsymbol{W}_r^{(\boldsymbol{C},1)}, \boldsymbol{h}_r^{(\boldsymbol{C},1)}, \boldsymbol{W}_r^{(\boldsymbol{C},2)}, \boldsymbol{h}_r^{(\boldsymbol{C},2)}\}$. This function always outputs a valid positive definite diagonal matrix, and lowers the computational cost of full covariance from $O(D^2K_b)$ to $O(DK_b)$. Similar parameterizations was used to parametrize a Gaussian distribution in [39]. The priors and means are also parametrized by two-layer NNs,

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{ir} &= f^{-1}_{\mu_r}(\boldsymbol{z}_i;\boldsymbol{M}_r^{(NN)}) \qquad (3)\\ &= \boldsymbol{W}_r^{(\boldsymbol{M},2)}\cdot\mathrm{Sigmoid}(\boldsymbol{W}_r^{(\boldsymbol{M},1)}\boldsymbol{z}_i + \boldsymbol{h}_r^{(\boldsymbol{M},1)}) + \boldsymbol{h}_r^{(\boldsymbol{M},2)},\\ \hat{\boldsymbol{\pi}}_i &= f^{-1}_{\pi}(\boldsymbol{w}_i;\boldsymbol{A}^{(NN)}) \qquad (4)\\ &= \mathrm{softmax}(\boldsymbol{W}^{(\boldsymbol{A},2)}\cdot\mathrm{ReLU}(\boldsymbol{W}^{(\boldsymbol{A},1)}\boldsymbol{w} + \boldsymbol{h}^{(\boldsymbol{A},1)}) + \boldsymbol{h}^{(\boldsymbol{A},2)}),\end{aligned}$$

where $\boldsymbol{M}_r^{(NN)} = \{\boldsymbol{W}_r^{(\boldsymbol{M},1)}, \boldsymbol{h}_r^{(\boldsymbol{M},1)}, \boldsymbol{W}_r^{(\boldsymbol{M},2)}, \boldsymbol{h}_r^{(\boldsymbol{M},2)}\}$ and $\boldsymbol{A}^{(NN)} = \{\boldsymbol{W}^{(\boldsymbol{A},1)}, \boldsymbol{h}^{(\boldsymbol{A},1)}, \boldsymbol{W}^{(\boldsymbol{A},2)}, \boldsymbol{h}^{(\boldsymbol{A},2)}\}$. There is no constraint for means so the output layer of $f^{-1}_{\mu_r}(\boldsymbol{z}_i;\boldsymbol{M}_r^{(NN)})$ is a linear function. Note that we parameterize priors as a vector $\hat{\boldsymbol{\pi}}_i = [\hat{\pi}_{i,1},\ldots,\hat{\pi}_{i,K_m}]$ using a network with a softmax as the output activation following the convention of neural networks that output a categorical distribution. We empirically observe that the softmax function with NN is not prone to unstable training, because the non-linear function better adapts to the softmax for predicting component probabilities, compared to the linear parametrization in (1).

3.2 Linear Hierarchical Latent Space

In (1), we use different latent variables to embed the prior, means and covariances to allow flexibility in representation. However, this treats the generation of each set of parameters independently. The dependencies among the prior, mean, and covariances is further modeled using a hierarchical latent space (HLS), which also reduces the dimensionality of the latent space (LS). In other words, we assume the

latent space $\{\boldsymbol{w}_i, \boldsymbol{y}_i, \boldsymbol{z}_i\}_{i=1}^N$ is generated from a HLS $\boldsymbol{v}$. The HLS can also be used to visualize the GMM manifold in a 2D or 3D space. Here we assume a linear relationship between HLS and LS. Define $\boldsymbol{L}_i = [\boldsymbol{w}_i^T, \boldsymbol{y}_i^T, \boldsymbol{z}_i^T]^T \in \mathbb{R}^{d_L}$ as the vector of concatenated latent variables with dimension $d_L = d_w + d_y + d_z$. Then, the latent vectors are generated from the HLS via $\boldsymbol{L}_i = \boldsymbol{H}^{(L)}\boldsymbol{v}_i$, where $\boldsymbol{v}_i \in \mathbb{R}^{d_v}$ are the HLS variables, d_v is the dimension of the HLS, and the matrix $\boldsymbol{H}^{(L)} \in \mathbb{R}^{d_L \times d_v}$ consists of basis vectors. Both $\boldsymbol{H}^{(L)}$ and $\{\boldsymbol{v}_i\}$ are estimated when learning the manifold.

3.3 Kernel Hierarchical Latent Space

The previous subsection proposes to learn a Linear HLS (LHLS) to capture the dependencies among priors, means and covariances, which could be quite limited when the generative process is highly nonlinear. We address this problem by introducing a kernel HLS (KHLS), i.e., the relationship between LS and HLS is defined by a kernelized function mapping $\boldsymbol{v}_i$ to $\boldsymbol{L}_i$. Define the positive definite kernel function between HLS vectors as $\kappa(\boldsymbol{v}_i, \boldsymbol{v}_j)$. The corresponding kernel matrix is $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ with entries $\kappa(\boldsymbol{v}_i, \boldsymbol{v}_j)$, and kernel vector is $\boldsymbol{k}_i \in \mathbb{R}^N$ between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ for all j. The kernelized function mapping is then

$$\boldsymbol{L}_i = \boldsymbol{H}^{(K)}(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}\boldsymbol{k}_i, \tag{5}$$

where λ is the regularization parameter, $\boldsymbol{I}$ is the identity matrix of size N, and $\boldsymbol{H}^{(K)} \in \mathbb{R}^{d_L \times N}$ is the collection of learned basis vectors. This formulation is inspired by multivariate kernel ridge regression [42] – in this context, the i-th column of $\boldsymbol{H}^{(K)}$ is equivalent to an "output" vector for the corresponding i-th "input" $\boldsymbol{v}_i$, which form an input/output training pair. However, in our case, both $\{\boldsymbol{v}_i\}$ and $\boldsymbol{H}^{(K)}$ are unknown and trained from the data by fitting the manifold.

An obvious shortcoming of this kernel formulation is its poor scalability: the computational cost of $(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}$ is $O(N^3)$. We introduce pseudo-points to overcome this problem, similar to [43] for improving the scalability of Gaussian processes. We define a set of N_p pseudo-points $\{\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_{N_p}\}$, where $\boldsymbol{\nu}_i \in \mathbb{R}^{d_v}$. The kernel matrix $\tilde{\boldsymbol{K}} \in \mathbb{R}^{N_p \times N_p}$ is now calculated between the N_p pseudo-points, and the kernel vector $\tilde{\boldsymbol{k}}_i \in \mathbb{R}^{N_p}$ is calculated between all pseudo-points and $\boldsymbol{v}_i$. The sparse kernelized function is then $\boldsymbol{L}_i = \boldsymbol{H}^{(K)}(\tilde{\boldsymbol{K}} + \lambda \boldsymbol{I})^{-1}\tilde{\boldsymbol{k}}_i$, where the basis vectors are $\boldsymbol{H}^{(K)} \in \mathbb{R}^{d_L \times N_p}$. In the sparse formulation, $\boldsymbol{H}^{(K)}$, $\{\boldsymbol{\nu}_i\}$, and $\{\boldsymbol{v}_i\}$ are estimated when learning the manifold. See Appendix D for a visualization of pseudo-points in the eye-fixation experiment. In the remainder of this paper, we only use the sparse formulation of KHLS.

3.4 Learning with EM Optimization

We next propose an algorithm for learning the reconstruction parameters and HLS variables from training data. Given a training set of N GMMs, let $p_i(\boldsymbol{x})$ be the distribution for the ith GMM with parameters $\{\pi_{ik}, \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}\}_{k=1}^{K_b}$. Denote the corresponding latent variables and HLS variables as $\{\boldsymbol{w}_i, \boldsymbol{z}_i, \boldsymbol{y}_i\}$ and $\{\boldsymbol{v}_i\}$, the reconstructed GMMs as $\{\hat{\pi}_{ir}, \hat{\boldsymbol{\mu}}_{ir}, \hat{\boldsymbol{\Sigma}}_{ir}\}_{r=1}^{K_m}$ with distribution $\hat{p}_i(\boldsymbol{x})$. The parameters $\boldsymbol{\Theta}^{(F)}$ in reconstruction functions $\{f_{\pi_r}^{-1}, f_{\mu_r}^{-1}, f_{\Sigma_r}^{-1}\}_{r=1}^{K_m}$ are

$$\boldsymbol{\Theta}^{(F)} = \{\boldsymbol{a}_r, \{\boldsymbol{m}_{rl}\}_{l=1}^{d_z}, \boldsymbol{b}_r, \{\boldsymbol{C}_{rl}\}_{l=1}^{d_y}, \beta_r\}_{r=1}^{K_m}, \tag{6}$$

for full covariance matrices, or

$$\boldsymbol{\Theta}^{(F)} = \{\boldsymbol{A}^{(NN)}, \{\boldsymbol{M}_r^{(NN)}, \boldsymbol{C}_r^{(NN)}\}_{r=1}^{K_m}\} \tag{7}$$

for diagonal covariance matrices. The parameters in HLS are $\boldsymbol{\Theta}^{(H)} = \boldsymbol{H}^{(L)}$ for linear HLS, or $\boldsymbol{\Theta}^{(H)} = \{\boldsymbol{H}^{(K)}, \boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_{N_p}\}$ for KHLS, and the HLS variables for the training data are $\boldsymbol{\Omega} = \{\boldsymbol{v}_i\}_{i=1}^N$.

The reconstruction and HLS parameters, and variables $\{\boldsymbol{\Theta}^{(F)}, \boldsymbol{\Theta}^{(H)}, \boldsymbol{\Omega}\}$ are obtained by minimizing the reconstruction loss between $p_i(\boldsymbol{x})$ and $\hat{p}_i(\boldsymbol{x})$, given by KLD [44],

$$\{\boldsymbol{\Theta}^{(F)^*}, \boldsymbol{\Theta}^{(H)^*}, \boldsymbol{\Omega}^*\} = \underset{\boldsymbol{\Theta}^{(F)}, \boldsymbol{\Theta}^{(H)}, \boldsymbol{\Omega}}{\operatorname{argmin}} \sum_{i=1}^N D_{KL}(p_i \| \hat{p}_i), \tag{8}$$

where $D_{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx$ is the KLD between p and q. Note that different combinations of parameterizations for $\boldsymbol{\Theta}^{(F)}$ are possible, as we explore in the experiments. We use a compact notation $\boldsymbol{\Theta}$ for all function parameters $\{\boldsymbol{\Theta}^{(F)}, \boldsymbol{\Theta}^{(H)}\}$. Decomposing the KLD and removing the first term, which is a constant w.r.t. $\{\boldsymbol{\Theta}, \boldsymbol{\Omega}\}$ yields an equivalent optimization problem based on the cross-entropy loss,

$$\begin{aligned} J_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}) &= -\sum_{i=1}^N \int p_i(\boldsymbol{x}) \log \hat{p}_i(\boldsymbol{x}|\boldsymbol{\Theta}, \boldsymbol{\Omega}_i) d\boldsymbol{x} \\ &= -\sum_{i=1}^N \int \sum_{k=1}^{K_b} \pi_{ik} \mathcal{N}_{ik}(\boldsymbol{x}) \log\{\sum_{r=1}^{K_m} \hat{\pi}_{ir} \hat{\mathcal{N}}_{ir}(\boldsymbol{x})\} d\boldsymbol{x}, \end{aligned} \tag{9}$$

where $\mathcal{N}_{ik}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})$ is the original Gaussian (kth component of the ith GMM) and $\hat{\mathcal{N}}_{ir}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\hat{\boldsymbol{\mu}}_{ir}, \hat{\boldsymbol{\Sigma}}_{ir})$ is the rth reconstruction component.

3.4.1 Variational Approximation

As the cross-entropy between two GMMs in (9) is intractable, we derive an approximation based on a variational upper-bound inspired by [14]. Introducing the variational parameters $\boldsymbol{q} = \{q_{kr}^{(i)}\}$, (9) is approximated as follows,

$$\begin{aligned} & J_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}) \\ &= -\sum_{i=1}^N \int \sum_{k=1}^{K_b} \pi_{ik} \mathcal{N}_{ik}(\boldsymbol{x}) \log\{\sum_{r=1}^{K_m} \frac{\hat{\pi}_{ir} \hat{\mathcal{N}}_{ir}(\boldsymbol{x})}{q_{kr}^{(i)}} q_{kr}^{(i)}\} d\boldsymbol{x} \\ &\leq -\sum_{i=1}^N \int \sum_{k=1}^{K_b} \pi_{ik} \mathcal{N}_{ik}(\boldsymbol{x}) \{\sum_{r=1}^{K_m} q_{kr}^{(i)} \log \frac{\hat{\pi}_{ir} \hat{\mathcal{N}}_{ir}(\boldsymbol{x})}{q_{kr}^{(i)}}\} d\boldsymbol{x} \\ &= -\sum_{i=1}^N \sum_{k=1}^{K_b} \pi_{ik} \sum_{r=1}^{K_m} q_{kr}^{(i)} \left[\log \frac{\hat{\pi}_{ir}}{q_{kr}^{(i)}} + \mathbb{E}_{\boldsymbol{x} \sim \mathcal{N}_{ik}}[\log \hat{\mathcal{N}}_{ir}(\boldsymbol{x})]\right] \\ &= \tilde{J}_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \boldsymbol{q}), \end{aligned} \tag{10}$$

where the inequality in (10) is based on Jensen's inequality. The variational parameter $q_{kr}^{(i)}$ can be interpreted as a soft assignment value for assigning the kth component of the ith GMM to the rth component of the reconstructed GMM.

3.4.2 Variational optimization algorithm

Using $\tilde{J}_{CE}$, we minimize an upper bound to J_{CE},

$$\{\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Omega}}, \hat{\boldsymbol{q}}\} = \underset{\boldsymbol{\Theta}, \boldsymbol{\Omega}, \boldsymbol{q}}{\operatorname{argmin}} \tilde{J}_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \boldsymbol{q}). \tag{11}$$

We adopt an alternating (variational EM) algorithm to solve the optimization problem:

(i) *Variational M-step*: Given $\hat{\boldsymbol{q}}$, optimize the manifold parameters and HLS variables: $\{\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Omega}}\} = \text{argmin}_{\boldsymbol{\Theta},\boldsymbol{\Omega}} \tilde{J}_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \hat{\boldsymbol{q}})$.

(ii) *Variational E-step*: Given $\{\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Omega}}\}$, calculate the optimal variational parameters: $\hat{\boldsymbol{q}} = \text{argmin}_{\boldsymbol{q}} \tilde{J}_{CE}(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Omega}}, \boldsymbol{q})$.

For (ii), the problem can be formulated as a constrained optimization problem,

$$\min_{\boldsymbol{q}} \tilde{J}_{CE}(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Omega}}, \boldsymbol{q}) \tag{12}$$

$$\text{s.t.} \sum_{r} q_{kr}^{(i)} = 1, \forall i, k \tag{13}$$

We use Lagrangian multiplier method to derive an analytical solution to $q_{kr}^{(i)}$ (see Appendix B for derivation),

$$\hat{q}_{kr}^{(i)} = \frac{\hat{\pi}_{ir}\hat{\mathcal{N}}_{ir}(\boldsymbol{\mu}_{ik})\exp\{-\frac{1}{2}\text{tr}(\hat{\boldsymbol{\Sigma}}_{ir}^{-1}\boldsymbol{\Sigma}_{ik})\}}{\sum_{n=1}^{K_m}\hat{\pi}_{in}\hat{\mathcal{N}}_{in}(\boldsymbol{\mu}_{ik})\exp\{-\frac{1}{2}\text{tr}(\hat{\boldsymbol{\Sigma}}_{in}^{-1}\boldsymbol{\Sigma}_{ik})\}}. \tag{14}$$

For (i), we use an alternating optimization strategy, i.e., optimizing one set of parameters while keeping others fixed. However, we employ different optimization methods for middle-size and large-scale datasets. Since full covariance matrices are applied in moderate-size datasets, we should optimize $\boldsymbol{\Theta}^{(F)} = \{\boldsymbol{a}_r, \{\boldsymbol{m}_{rl}\}_{l=1}^{d_z}, \{\boldsymbol{C}_{rl}\}_{l=1}^{d_y}, \boldsymbol{b}_r, \beta_r\}_{r=1}^{K_m}$ and HLS variables. In this setting, fast solvers can be derived for sub-problems. For example, $\boldsymbol{m}_{rl}$ and $\boldsymbol{b}_r$ is obtain in closed-form, $\boldsymbol{C}_{rl}\boldsymbol{C}_{rl}^T$ is solved by semidefinite programming, $\boldsymbol{a}_r$ and β_r are solved by the Newton-Raphson method (see Appendix C). For large-scale datasets, we use Tensorflow to implement the gradient descent algorithm so that we can make use of automatic differentiation and use GPUs to accelerate the optimization, which allows more flexible parameterizations and ameliorates scalability.

One drawback of this learning algorithm is that the assignment variables $\hat{\boldsymbol{q}}$ affect the training result to a great extent, and poor initialization may cause the optimizer to become stuck in a local minimum. To help $\hat{\boldsymbol{q}}$ escape from local minima, after each iteration, we use a Metropolis-Hasting (MH) sampler for $\hat{\boldsymbol{q}}$ [24, 25], which randomly swaps assignments for a random Gaussian component (more details are provided in Appendix C). The learning algorithm is summarized in Algorithm 1.

3.4.3 Regularization

In (1), the HLS variables and manifold parameters are unconstrained, and thus multiple equivalent solutions exist by scaling the latent variables and principal axes in opposite directions. The similar argument applies to (2). To remove this ambiguity, we apply regularization on latent variables, which effectively constrains the principal functions, similar to the constraint on scales of principal vectors in PCA,

$$\rho_1(\boldsymbol{\Omega}) = \sum_{i=1}^{N} c_w\|\boldsymbol{w}_i\|^2 + c_z\|\boldsymbol{z}_i\|^2 + c_y\|\boldsymbol{y}_i\|^2 + c_v\|\boldsymbol{v}_i\|^2, \tag{15}$$

where c_w, c_z, c_y, c_v are the regularization hyperparameters.

A second regularization is used to constrain the distance between the HLS variables so that the low-dimensional latent space preserves the distance information on the high-dimensional GMM manifold. Specifically, we require the distance of any two GMMs in HLS matches their symmetric KL divergence $D_{SKL}(p_i, p_j) = D_{KL}(p_i, p_j) + D_{KL}(p_j, p_i)$,

$$\rho_2(\boldsymbol{\Omega}) = c_{KL}\sum_{i,j}^{N}(\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 - D_{SKL}(p_i, p_j))^2. \tag{16}$$

Note that the symmetric KL is only computed once before the optimization.

To better condition the assignment variables $q_{kr}^{(j)}$ and prevent degeneration to uniform assignments, following previous work [14, 15], we introduce virtual samples where the variables $\boldsymbol{x}$ are replicated with i.i.d. distributions, i.e. $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_v}\}$. Now the objective function is

$$\begin{aligned} J_{CE}^{(vs)}(\boldsymbol{\Theta}, \boldsymbol{\Omega}) &= -\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{X}\sim p_i}[\log\hat{p}_i(\boldsymbol{X}|\boldsymbol{\Theta}, \boldsymbol{\Omega})] \\ &= -\sum_{i=1}^{N}\sum_{k=1}^{K_b}\pi_{ik}\mathbb{E}_{\boldsymbol{X}\sim p_{ik}}[\log\hat{p}_i(\boldsymbol{X}|\boldsymbol{\Theta}, \boldsymbol{\Omega})] \\ &\leq -\sum_{i=1}^{N}\sum_{k=1}^{K_b}\pi_{ik}\sum_{r=1}^{K_m}q_{kr}^{(i)}\{\log\frac{\hat{\pi}_{ir}}{q_{kr}^{(i)}} + \mathbb{E}_{\boldsymbol{X}\sim p_{ik}}[\log\hat{p}_{ir}(\boldsymbol{X}|\boldsymbol{\Theta}, \boldsymbol{\Omega})]\}. \end{aligned} \tag{17}$$

The virtual samples are i.i.d., and thus the expectation term $\mathbb{E}_{\boldsymbol{X}\sim p_{ik}}[\log\hat{p}_{ir}(\boldsymbol{X}|\boldsymbol{\Theta}, \boldsymbol{\Omega})]$ can be written as $N_v\mathbb{E}_{\boldsymbol{x}\sim p_{ik}}[\log\hat{p}_{ir}(x|\boldsymbol{\Theta}, \boldsymbol{\Omega})]$. The upper bound is then

$$\begin{aligned} &\tilde{J}_{CE}^{(vs)}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \boldsymbol{q}) = \\ &-\sum_{i=1}^{N}\sum_{k=1}^{K_b}\pi_{ik}\sum_{r=1}^{K_m}q_{kr}^{(i)}\{\log\frac{\hat{\pi}_{ir}}{q_{kr}^{(i)}} + N_v\mathbb{E}_{\boldsymbol{x}\sim p_{ik}}[\log\hat{p}_{ir}(\boldsymbol{x}|\boldsymbol{\Theta}, \boldsymbol{\Omega})]\} \end{aligned} \tag{18}$$

Using similar techniques in (14), the optimal variational parameters with virtual samples are,

$$\hat{q}_{kr}^{(i)} = \frac{\hat{\pi}_{ir}\hat{\mathcal{N}}_{ik}(\boldsymbol{\mu}_{ik})^{N_v}\exp\{-\frac{1}{2}N_v\text{tr}(\hat{\boldsymbol{\Sigma}}_{ir}^{-1}\boldsymbol{\Sigma}_{ik})\}}{\sum_{n=1}^{K_m}\hat{\pi}_{in}\hat{\mathcal{N}}_{in}(\boldsymbol{\mu}_{ik})^{N_v}\exp\{-\frac{1}{2}N_v\text{tr}(\hat{\boldsymbol{\Sigma}}_{in}^{-1}\boldsymbol{\Sigma}_{ik})\}}. \tag{19}$$

This expression is similar to deterministic annealing [45], derived from the maximum entropy principle to avoid poor

Algorithm 1 Optimization Algorithm for PRIMAL

Input: $\{\pi_{ik}, \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}\}_{k=1}^{K_b}, i = 1, \ldots, N$
Parameters: N_v, c_w, c_z, c_y, c_{KL}, c_v, T_{MH}
Output: $\{\hat{\boldsymbol{q}}, \hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Omega}}\}$

```
Initialize {q, Θ, Ω}
while not converge do
    M-step: Fix q̂ and optimize {Θ, Ω}
    E-step: Fix {Θ̂, Ω̂} and optimize q̂, save J̃_CE
    Randomly sample a GMM, denoted as i_s
    Randomize variational parameters of all component of i_s th GMM as q̂^(i_s)* and calculate J̃*_CE using q̂^(i_s)*
    if p_swap = exp(−J̃*_CE/T_MH) / exp(−J̃_CE/T_MH) > 1 then
        Replace q̂^(i_s) with q̂^(i_s)*
    else
        Do the above replacement with probability p_swap
    end if
end while
return {q̂, Θ̂, Ω̂}
```

local optima. To see this, we add a regularization term consisting of negative entropy of $\boldsymbol{q}$ to our objective function,

$$\min_{\boldsymbol{\Theta},\boldsymbol{\Omega},\boldsymbol{q}} \tilde{J}_{CE}(\boldsymbol{\Theta},\boldsymbol{\Omega},\boldsymbol{q}) + T_{DA}\sum_{i=1}^{N}\sum_{k=1}^{K_b}\pi_{ik}\sum_{r=1}^{K_m} q_{kr}^{(i)}\log q_{kr}^{(i)}, \quad (20)$$

where T_{DA} is the temperature parameter that affects the randomness of the variational parameters. Solving this regularized problem will give a similar solution for $\boldsymbol{q}$ as (19), with N_v replaced with $\frac{1}{T_{DA}+1}$. This provides insights to the effect of the virtual samples: more virtual samples lead to more deterministic variational parameters. Thus we can adapt the number of virtual samples to change the homogeneity of the variational parameters. When N_v is 0, the variational parameters will be uniform, and when N_v is large, variational parameters will take binary values (0 or 1).

Finally, the optimization with regularization is

$$\{\hat{\boldsymbol{\Theta}},\hat{\boldsymbol{\Omega}},\hat{\boldsymbol{q}}\} = \operatorname*{argmin}_{\boldsymbol{\Theta},\boldsymbol{\Omega},\boldsymbol{q}} \tilde{J}_{CE}^{(vs)}(\boldsymbol{\Theta},\boldsymbol{\Omega},\boldsymbol{q}) + \rho_1(\boldsymbol{\Omega}) + \rho_2(\boldsymbol{\Omega}). \quad (21)$$

3.5 Inference

After learning the manifold $\hat{\boldsymbol{\Theta}}$, a novel GMM is embedded in the manifold by minimizing the cross-entropy between the novel GMM $\{\pi_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\}_{k=1}^{K}$ and its reconstruction,

$$\hat{\boldsymbol{v}} = \operatorname*{argmin}_{\boldsymbol{v}} -\int\sum_{k=1}^{K_b}\pi_k\mathcal{N}_k(\boldsymbol{x})\log\sum_{r=1}^{K_m}\hat{\pi}_r\hat{\mathcal{N}}_r(\boldsymbol{x})d\boldsymbol{x} + c_v\|\boldsymbol{v}\|^2 \quad (22)$$

where the distribution $\hat{\mathcal{N}}_r(\boldsymbol{x})$ is a function of $(\boldsymbol{w},\boldsymbol{z},\boldsymbol{y})$. The optimization problem is solved using the same algorithm in Section 3.4, but keeping the manifold parameters $\hat{\boldsymbol{\Theta}}$ fixed. This is equivalent to defining an implicit forward mapping f from the distribution $p(\boldsymbol{x})$ to the latent space $(\boldsymbol{w},\boldsymbol{z},\boldsymbol{y})$. Note that several novel GMMs can be input for inference by optimizing several HLS variables in a parallel manner.

3.6 Connection to Generalized Linear Models

The hierarchical latent space formulation can be better understood from the perspective of Generalized Linear Models (GLM) [46]. A GLM is made up of a linear predictor $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$ and a link function $\boldsymbol{\eta} = g(\boldsymbol{\zeta})$, where $\boldsymbol{\zeta}$ are parameters of the output distribution $\boldsymbol{Y}$. The linear predictor is the systematic component that embodies our belief of the linear relationship between the input $\boldsymbol{X}$ and latent variables $\boldsymbol{\eta}$. The link function models the function between latent variables $\boldsymbol{\eta}$ and expected value of target variables $\boldsymbol{Y}$. A classical linear regression model can be regarded as a special case of GLMs in which the link function is an identity mapping and the output distribution is a Gaussian. Hence, GLMs enrich the hypothesis space of linear predictors by introducing a possibly nonlinear link function and other output distributions. GLM has also been kernelized by assuming a kernel function for the predictor $\boldsymbol{\eta}(\boldsymbol{X})$ [47].

PRIMAL can be interpreted as a type of generalized linear/kernel model. PRIMAL learns a manifold for a set of GMMs assuming a generative process from hierarchical latent variables to GMM parameters: $\boldsymbol{v} \xrightarrow{h^{-1}} (\boldsymbol{w},\boldsymbol{y},\boldsymbol{z}) \xrightarrow{f^{-1}} (\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})$. The first generative process from hierarchical latent variables to latent variables h^{-1}, which models the relationship between "input" variables $\boldsymbol{v}$ and latent variables $(\boldsymbol{w},\boldsymbol{y},\boldsymbol{z})$, is equivalent to the predictor or systematic component in GLMs. The difference is that GLM assumes a supervised setting (using both inputs $\boldsymbol{X}$ and outputs $\boldsymbol{Y}$), while PRIMAL uses an unsupervised setting (only output distributions $(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})$ are known, and the "inputs" $\boldsymbol{v}$ of the HLS are inferred). The second generative process from latent variables to GMM parameters f^{-1} is equivalent to the link function in GLMs because it relates latent variables to parameters of the target variables. We propose special link functions for GMM parameters to satisfy their constraints. GLMs are learned via maximum likelihood estimation on a finite set of samples. PRIMAL extends GLM to learning through infinite samples (probability distributions) by minimizing the KLD between probability distributions.

4 Experiments

We conduct experiments with PRIMAL on learning GMM manifolds on a variety of synthetic and real datasets: 1) synthetic GMMs generated from a ground-truth GMM manifold; 2) GMMs of eye-fixation data from young and old subjects [20]; 3) GMMs of flow cytometry data from acute myeloid leukemia (AML) positive and healthy donor patients [19]; 4) topic models in the form of GMMs from BBC News dataset [48].

4.1 Evaluation Metrics

We first describe how to evaluate the PRIMAL (and related methods). Given a set of GMMs, PRIMAL is used to estimate a low-dimensional GMM manifold represented by the HLS. We evaluate the learned GMM manifold and HLS in 4 ways:

1) KLD reconstruction loss on held-out test GMMs, which measures how well the learned manifold represents the true underlying manifold.
2) Correlation between the HLS and other dependent variables (metadata), which measures whether the HLS dimensions are meaningful w.r.t. other measured data.
3) Classification accuracy using latent discriminant analysis (LDA) in the latent space, which measures the correlation between the HLS and data labels, as well as the efficacy of HLS in downstream tasks. We use LDA to learn a 1D discriminant space from the trained HLS variables, then map the test HLS variables to the same space and use k Nearest Neighbors (kNN) to perform the classification. If not specified, the k is set as 1.
4) Visualization of the GMM manifold, which provides better understanding of the data and also highlights the interpretability of our method.

4.2 Experiment Setup

In the synthetic data experiment, we generate the input GMMs from a given parametric GMM manifold. The test GMMs are generated from points interpolated between the training points on the manifold. In real-world applications, we randomly sample subjects as the training/testing set and then use EM algorithm to estimate GMM for each subject's sample collection. See the following sections for details of data processing.

Given input GMMs, we use Algorithm 1 to learn the PRIMAL-GMM manifold. We denote PRIMAL using a linear HLS as PRIMAL-L, and using the non-linear kernel HLS as PRIMAL-NL. The dimension of HLS and LS in linear

Dataset	Metric	KPCA	GPLVM	FINE	PRIMAL(L)	PRIMAL(NL)
Synthetic (L)	KL Loss (Train)	352.3	3.656	-	**3.474e-2**	-
	KL Loss (Test)	358.1	4.678	-	**5.419e-2**	-
Synthetic (NL)	KL Loss (Train)	508.0	**9.236e-4**	-	-	5.546e-2
	KL Loss (Test)	714.8	**2.375e-2**	-	-	4.844e-2
Eye Fixations	KL Loss	1.749	0.8257	-	0.7100	**0.5011**
	LDA Acc	51.5%	51.5%	45.5%	81.8%	**97.0%**
Flow Cyto (M6)	KL Loss	4.794	4.100	-	2.903	**2.226**
	LDA Acc	95.00%	85.00%	75.00%	90.00%	**95.00%**
Topic Model	KL Loss	38.97	11.13	-	**9.183**	10.23
	LDA Acc ($k_{NN} = 16$)	55.00%	58.33%	71.67%	**78.33%**	66.67%

TABLE 1: KL reconstruction loss for held-out test GMMs and LDA classification accuracy in the latent space. The LDA accuracy is the result of the smallest KL loss among several trials. Bold text denotes the best performance.

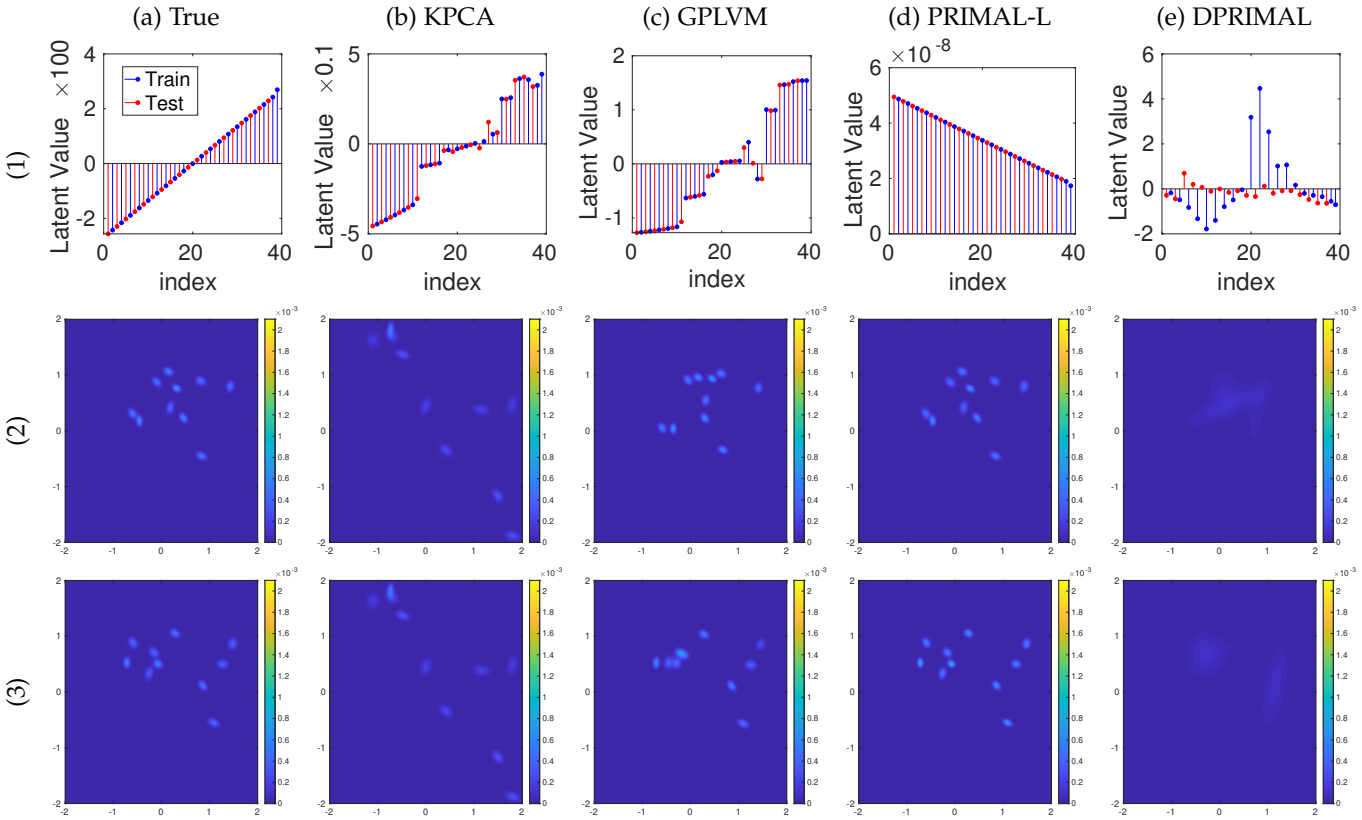


Fig. 2: Experiment on Synthetic Data. The positions of HLS variables are kept the same as that of ground truth. PRIMAL-L learns a smooth latent structure and achieve the best reconstruction.

synthetic data is set as $d_v = 1, d_w = 1, d_y = d_z = 2$. We set $d_v = 3, d_w = d_y = d_z = 2$ for all other experiments. The hyperparameter settings for PRIMAL-L/NL are shown in Appendix D. For the kernel in KHLS, we use a combination of Gaussian kernel and polynomial kernel,

$$\kappa(\boldsymbol{v}_i, \boldsymbol{v}_j) = \sigma_{es}^2 \exp\left(-\frac{\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2}{2\sigma_w^2}\right) + \sigma_{ps}^2 \boldsymbol{v}_i^T \boldsymbol{v}_j. \quad (23)$$

We compare PRIMAL with GPLVM [37], KPCA [32] and FINE [18]. GPLVM and KPCA cannot directly take GMMs as input so we convert GMMs into vectors. To vectorize a GMM, we first transform its parameters so that valid GMMs can be obtained in the reconstruction stage: the prior is mapped to an unconstrained space using the inverse softmax function, the covariance matrix is represented by its Cholesky decomposition. The transformed parameters of the GMM are then concatenated into a long vector. However, caution should be exercised in the concatenation because the ordering of Gaussian components will affect the vectors and the final embeddings. We normalize the order as follows. For all GMMs in the dataset, the parameters for each Gaussian component are converted into vectors and then mapped to a 1-D line using PCA. The PCA coefficient for each Gaussian component then determines the order during concatenation. Note that PRIMAL is *invariant* to the component order in that variational parameters $\boldsymbol{q}$ will determine the assignment between input components to reconstructed components.

In the original paper [18], FINE was formulated to take a single Gaussian as input. Here we extend FINE to embed GMMs by approximating the KLD between GMMs using the variational approximation in [49]. Note that FINE cannot reconstruct GMMs from the latent space because it is based on MDS, which directly optimizes the latent variables according to the distance matrix.

Our framework uses a two-stage estimation procedure: 1) subsets of data (e.g., corresponding to subjects) are summarized using GMMs; 2) the GMM manifold is estimated from the individual GMMs. One reasonable alternative to our framework is to directly learn the GMM manifold from the data samples, denoted as Direct-PRIMAL (DPRIMAL). After learning the latent space $\{\boldsymbol{w}, \boldsymbol{y}, \boldsymbol{z}\}$ with DPRIMAL, we

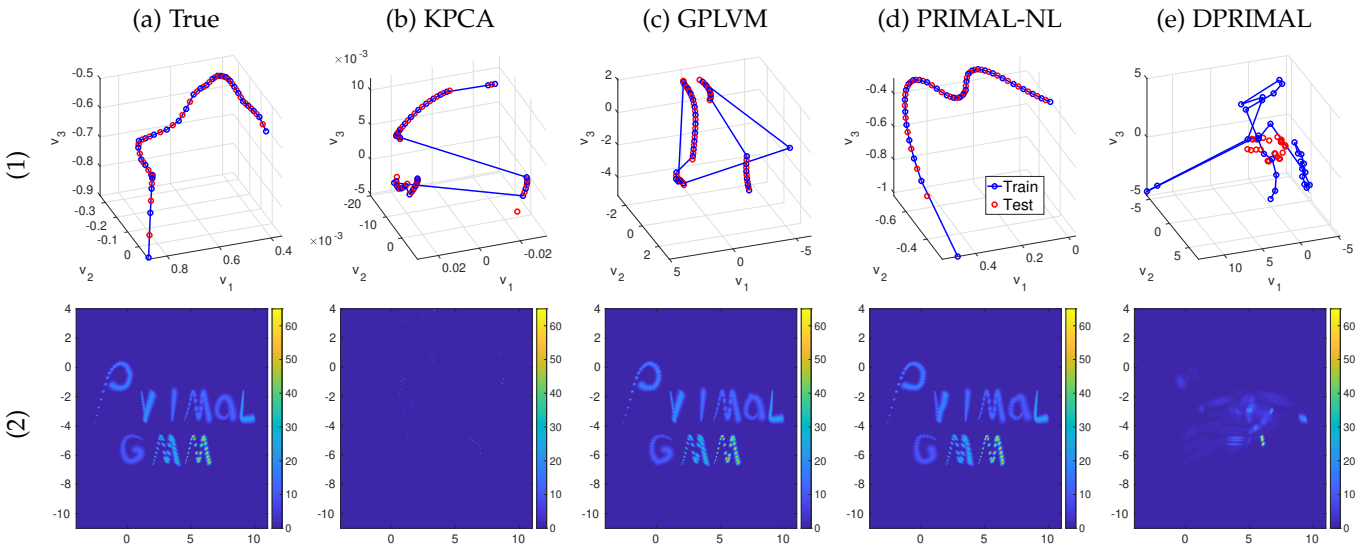


Fig. 3: Experiment on Synthetic Data. PRIMAL-NL learns a smooth latent structure and achieves a good reconstruction.

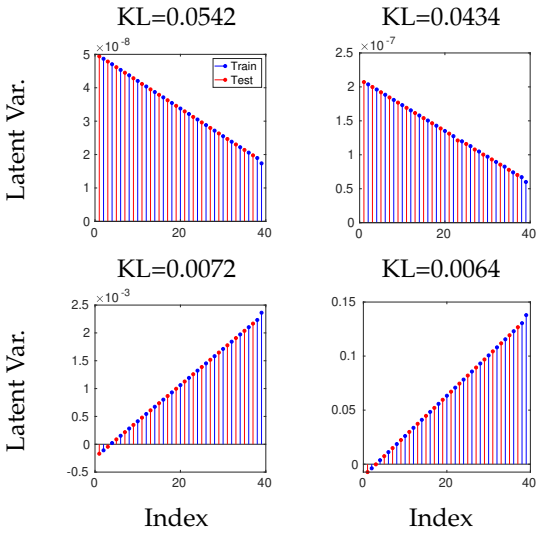


Fig. 4: HLS of PRIMAL-L under different hyperparameter settings (c_{wyz}, c_v): (1,0.2), (0.1,0.1), (0.01,0.01), (0.001,0.001); from left to right, then top to bottom.

run PCA on the latent space to obtain a hierarchical space analogous to HLS.

4.3 Synthetic Data

We use two syntheic datasets of GMMs generated by a linear and a nonlinear latent function to test PRIMAL with a linear HLS and a kernel HLS, respectively.

The linear synthetic data consist of 39 synthetic GMMs with 10 components, whose means and covariances are generated from a 1D latent space, according to known latent functions (Fig. 2-a1), and the priors are uniform. 20 GMMs are used for training and the other 19 GMMs are reserved for testing. We use full-covariance parameterization $\boldsymbol{\Theta}^{(fc)} = \{\boldsymbol{a}_r, \{\boldsymbol{m}_{rl}\}_{l=1}^{d_z}, \{\boldsymbol{C}_{rl}\}_{l=1}^{d_y}, \boldsymbol{b}_r, \beta_r\}_{r=1}^{K_m}$ for PRIMAL-L. Fig. 2 shows results of PRIMAL with a linear HLS (PRIMAL-L), GPLVM and KPCA. PRIMAL-L learns both a smooth latent space and achieves the best reconstruction error (Table 1), compared with other methods.[2] KPCA neither reconstructs GMMs nor learns a smooth latent space. GPLVM reconstructs some of the GMMs but fails on others, and the latent space is not consistent with the ground truth.

We create a more challenging synthetic dataset by generating GMMs from a 3D space by some nonlinear function (Figure 3-a). There are 30 GMMs (blue points in HLS) for training and 29 GMMs (red points in HLS) for testing (see GMMs in Fig. 3-a2). Each GMM has 9 components, where each component corresponds to a letter in "PRIMALGMM" and its position in the letter is mapped from the HLS. We use neural networks as the parameterization of means in PRIMAL-NL, i.e., $\boldsymbol{\Theta}^{(mNN)} = \{\boldsymbol{a}_r, \boldsymbol{M}_r^{(NN)}, \{\boldsymbol{C}_{rl}\}_{l=1}^{d_y}, \beta_r\}_{r=1}^{K_m}$, and the hyperparameters for the KHLS kernel are $\sigma_{es} = \sigma_{ps} = \sigma_w = 1$. Both PRIMAL-NL and GPLVM reconstruct the GMM density well while KPCA fails to reconstruct the density. In terms of the latent space, only PRIMAL-NL learns a smooth latent space. Table 1 shows a large gap between training and testing reconstruction error of GPLVM, which indicates it is overfitting to the training data. In contrast, PRIMAL-NL has similar training and testing loss in terms of density reconstruction.

Finally, we use DPRIMAL to learn a latent space $\{\boldsymbol{w}, \boldsymbol{y}, \boldsymbol{z}\}$ for both linearly and nonlinearly synthetic data, and use PCA to obtain the 1 or 3 principal components for latent space as an analogy to our linear or nonlinear HLS. The mapping functions from $\{\boldsymbol{w}, \boldsymbol{y}, \boldsymbol{z}\}$ to GMM parameters use the full-covariance parameterization. Fig. 2(e) and Fig. 3(e) show the learned principal components and suggest that DPRIMAL neither learns a good GMM manifold (average KLD loss of 2.749 and 3.909 for the linear and nonlinear cases) compared to PRIMAL, nor learns the correct latent space.

4.3.1 *Sensitivity Analysis*

We investigate the sensitivity of our algorithm to the regularization hyperparameters (c_{wyz}, c_v), where (c_w, c_y, c_z)

2. Note that PRIMAL-L has multiple equivalent solutions by multiplying a column of $\boldsymbol{H}$ and corresponding entries in $\boldsymbol{v}$, as in standard PCA. Thus, the signs of the HLS values may be flipped, as in Fig. 2d-1. Nonetheless, the interpretability of the HLS is not affected, since flipping signs only affects the correlation direction (positive vs. negative), but not the statistical significance.

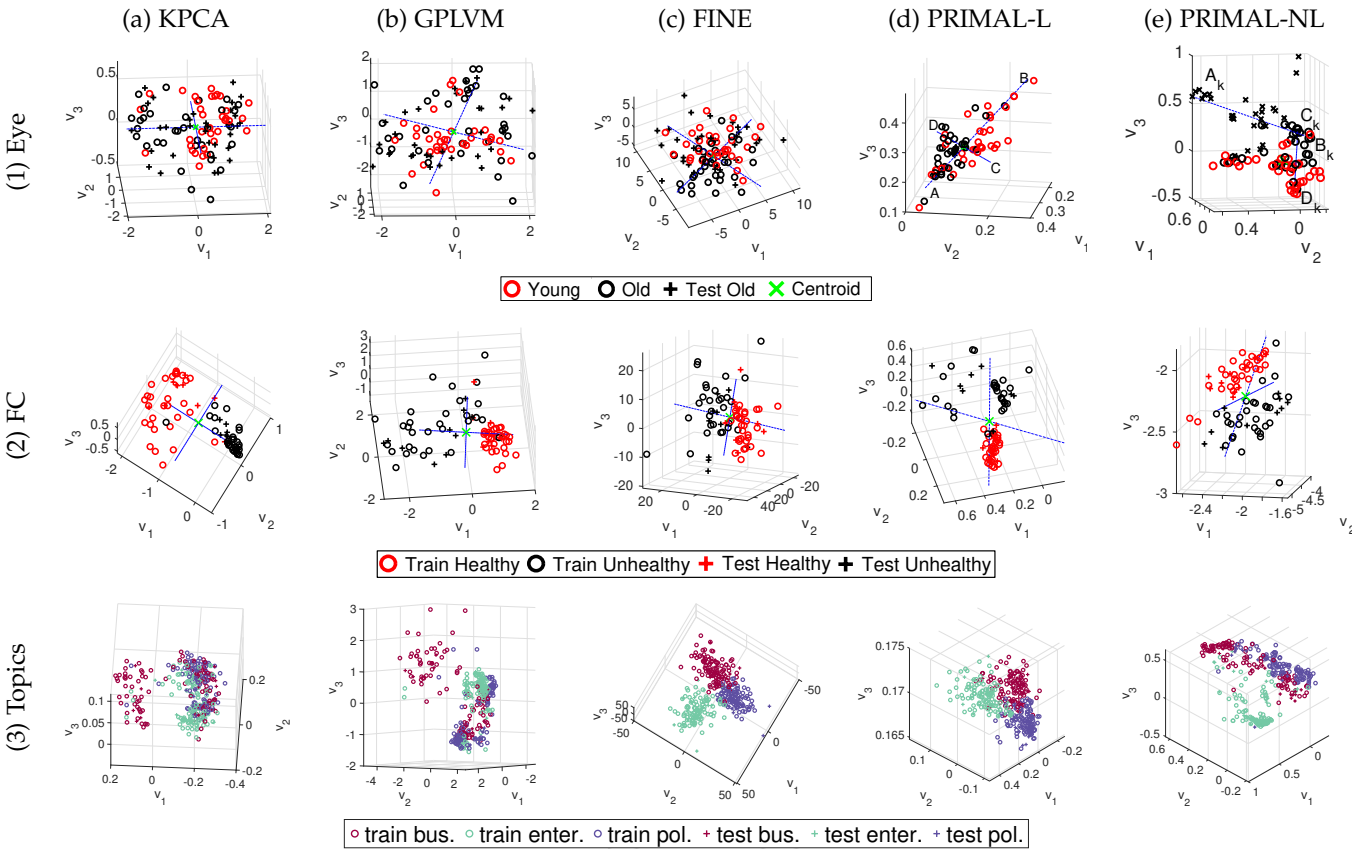


Fig. 5: Hiearchical Latent Space Visualization. (1) Eye-fixation data; (2) Flow Cytometry data (FC); (3) Topic models. The 2D view of the 3D axis is selected to best show the class separation in the HLS.

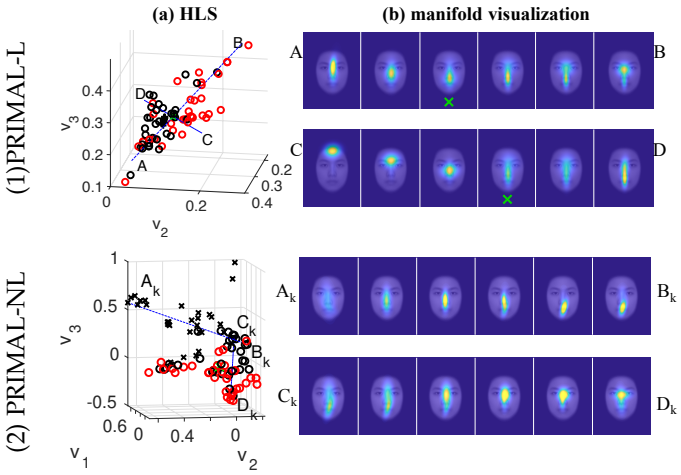


Fig. 6: Manifold Visualization of Eye-Fixation Data of PRIMAL.

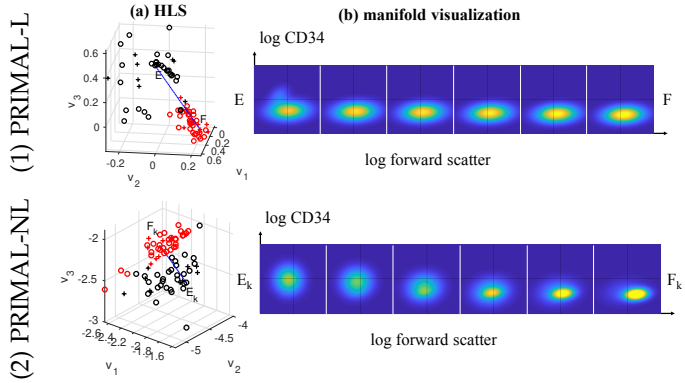


Fig. 7: Manifold Visualization of Flow Cytometry Data of PRIMAL.

share the same value c_{wyz}. In Fig. 4, we show the result of PRIMAL-L on the linear synthetic data under 4 hyperparameter settings. PRIMAL-L finds the true latent structure of data and achieves good reconstruction error under all hyperparameter settings, indicating that PRIMAL is robust to the hyperparameter setting in this range. When the regularization becomes weaker (smaller hyperparameter values), PRIMAL-L achieves smaller KL loss and the scale of the latent variables becomes larger. However, the optimization failed when we set (c_{wyz}, c_v) smaller than 0.001, due to infinite values in the latent variables. This phenomenon is reasonable, since PCA also requires a unit-norm constraint on the principal vectors to avoid arbitrary scale in the latent variables. Thus, we note that relatively large-valued hyperparameters are essential to PRIMAL, but as long as the optimization does not collapse, the hyperparameter values will not affect the result much.

4.4 Eye-Fixation Data

Eye-fixation data consists of a time series of 2D location points of eye fixations of a human subject looking at image stimuli. Previous work [50] has shown that a subject's eye fixation patterns during a face recognition task can be clustered into two general strategies: a holistic strategy that looks at the face center, and an analytic strategy that looks at the eyes and mouth. These patterns have also been shown to be correlated with the cognitive ability of older adults [20].

In this experiment, we use a dataset containing eye-fixation recordings of 34 young people and 67 old ones (34 old adults for training, 33 for testing). We model each person's eye-fixation pattern with a GMM, where each

component corresponds to one region-of-interest (ROI) on a face. As suggested by [20], we use $K_b = 3$ components corresponding to 3 ROIs. We use both PRIMAL-L and PRIMAL-NL to learn GMM manifolds for eye-fixation data and their parameterizations are $\mathbf{\Theta}^{(fc)}$ and $\mathbf{\Theta}^{(pmNN)} = \{\boldsymbol{A}^{NN}, \{\boldsymbol{M}_r^{(NN)}, \{\boldsymbol{C}_{rl}\}_{l=1}^{d_y}, \beta_r\}_{r=1}^{K_m}\}$ respectively.

	PRIMAL-L	PRIMAL-NL	GPLVM	FINE	KPCA
p	**<0.0001**	**<0.0001**	0.1586	**0.0001**	**0.0157**
R^2	**0.3180**	0.2938	0.0773	0.2703	0.1485

TABLE 2: Eye fixation data: correlation between the HLS and age using multivariate linear regression analysis.

The HLS is shown in Figure 5-1. Only in the HLS of PRIMAL-L and PRIMAL-NL can we observe that there are different regions for older (AD quadrant in Fig. 5-1a and B_kD_k quadrant in Fig. 5-1b) and young (CD quadrant in Fig. 5-1a and A_kC_k quadrant in Fig. 5-1b) adults, and the test old adults' GMMs (black plus points) are all embedded into the older regions. In contrast, the other three methods embed testing GMMs into their latent spaces in an undesirable way (see LDA accuracy for "Eye Fixations" in Table 1). We examine the correlation between hierarchical latent variables and ages of subjects using the multivariate linear regression analysis and show the p-value and R^2 value in Table 2. The result indicates that HLS variables of PRIMAL are correlated with ages at a statistically significant level and have the largest R^2 statistic.

We visualize the reconstructed GMMs of PRIMAL-L and PRIMAL-NL from their respective HLS's in Fig. 6. Along $\overline{AB}$ in the HLS, from the centroid towards A shows a vertically shaped ROI going up towards the upper center of the face, and towards B shows a vertically shaped ROI at the nose and a more horizontally shaped ROI around the eyes. Along $\overline{CD}$, from the centroid to C shows a horizontally shaped ROI going upwards, whereas towards D shows a vertically shaped ROI going downwards. As older adults' ROIs focus on the face midline, their ROIs are vertically shaped, and thus they are embedded into the AD quadrant. In contrast, young adults look around the eye regions and have horizontal ROIs around the eyes, and thus they are embedded into the BC quadrant. In the non-linear HLS along $\overline{C_kD_k}$, a vertically shaped ROI at the mouth and nose changes to a horizontal ROI around eyes. Along $\overline{A_kB_k}$, an ROI around the nose changes to one that is around the mouth. This finding is consistent with previous work [20], but PRIMAL is able to visualize the continuous change of eye gaze strategy, instead of only discrete clusters as in [20].

4.5 Flow Cytometry Data

Flow cytometry data measures cell properties of patients for medical diagnosis. Cells are dyed with fluorescent markers, and then forced to pass through a laser beam one at a time in the flow cytometry instrument. The detector records the light scattering of the cells and the emitted light of the fluorescent markers, which reflect certain properties of the cells like sizes and surface proteins. The dimensionality of flow cytometry data samples ranges from 5-8, and the number of samples for each patient is often thousands, which makes direct analysis cumbersome. We propose to model each patient's data as a GMM and embed them into a low-dimensional space. Here we use an open AML dataset [19] with 7-dim features and each patient is labeled

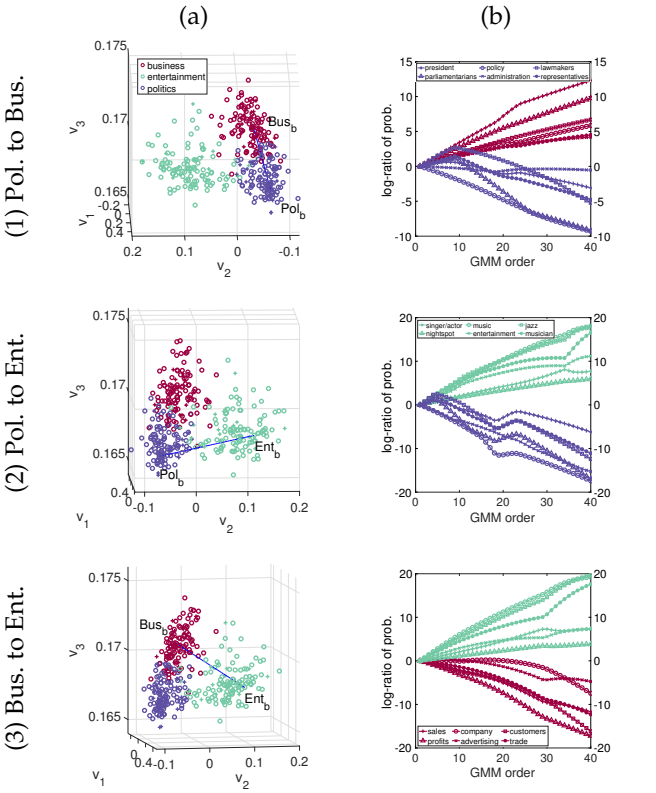


Fig. 8: Manifold visualization of topic model HLS learned by PRIMAL-L. Colors denote topics while markers denote different words. In (b), word samples of each topic keep the same and from GMM order 0 to 40 denotes from start to end of the visualization line.

as either healthy or unhealthy. To better interpret our result, we choose the tube containing marker CD34, which is the same visualized marker in [19]. 30 healthy and 30 unhealthy patients are used for training, and another 10 healthy and 10 unhealthy are used for testing. We run EM on each patient's data using $K \in \{2, 3, 4, 5\}$ and find that when $K > 2$ there are many subjects ($> 80\%$) with low-weight components. Hence, we use GMMs with 2 components to model each patient. We use a smaller scale parameter for the exponential kernel, i.e., $\sigma_{es} = 0.1$, because we find that the nonlinearity will harm the KL loss. The parameterization for PRIMAL-L is $\mathbf{\Theta}^{(fc)}$ and for PRIMAL-NL is $\mathbf{\Theta}^{(mNN)}$.

Fig. 5-2 shows latent spaces learned by various methods. Although all methods show a good qualitative result, quantitative assessments in Table 1 show that PRIMAL-L/NL achieves the best performance in terms of LDA accuracy and KL reconstruction loss. In Fig. 7, we visualize the GMM change along the direction from unhealthy patients to healthy patients in HLS of PRIMAL-L/NL. To compare with previous work [19], we visualize two dimensions, log CD34 and log forward scatter. The value of the CD34 marker decreases when moving from the unhealthy to the healthy region, which is consistent with the previous work on classification [19]. Our method learns an interpretable HLS, on which classification can also be performed.

4.6 Topic Models

Previous work has proposed to model documents as GMMs to represent both word semantics and topic diversity [12]. The words are modeled as high-dimensional vectors in a word space, and the topics are Gaussian distributions over

Politics →	1. A group of MPs has tried to raise the pressure on Tony Blair over reform to the House of Lords by publishing a detailed blueprint for change. 2. Lib Dem leader Charles Kennedy has said voters now have a "fundamental lack of trust" of Tony Blair as prime minister. 3. Tony Blair faces his first prime minister's questions of 2005 after a week of renewed speculation about his relationship with Gordon Brown. 4. Ministers would not rule out scrapping the Child Support Agency if it failed to improve, Work and Pensions Secretary Alan Johnson has warned. 5. First Minister Jack McConnell has ordered a report on the decision to allow a paranoid schizophrenic knife attacker to go on a visit unguarded. 6. He called it his "masochism strategy" in the run-up to the Iraq war and now Tony Blair has signed up for another dose of pain. 7. For the umpteenth time, Tony Blair and Gordon Brown are said to have declared all out war on each other. 8. Venezuela is to review all foreign investment in its mining industries in an effort to strengthen its indigenous industrial output. 9. An ex-chief financial officer at Boeing has received a four-month jail sentence and a fine of $250,000 (£131,961) for illegally hiring a top Air Force aide. 10. The US government is to investigate two airlines-US Airways and Delta Air Lines' Comair subsidiary 11. As European leaders gather in Rome on Friday to sign the new EU constitution, many companies will be focusing on matters much closer to home. 12. Pan-European stock market Euronext has approached the London Stock Exchange (LSE) about a possible takeover bid.
← Business →	1. Pan-European stock market Euronext has approached the London Stock Exchange about a possible takeover bid. 2. US mortgage company Fannie Mae should restate its earnings, a move that is likely to put a billion-dollar dent in its accounts. 3. UK advertising giant WPP has posted larger-than-expected annual profits and predicted that it will outperform the market in 2005. 4. Swiss cement firm Holcim has bid $800m (£429m) to buy two Indian cement firms and a holding company in the country. 5. Fans who buy tickets for this year's Glastonbury festival will be issued with photo ID cards in an attempt to beat touts. 6. Brad Pitt, Robert De Niro and Hugh Grant have been added to the line-up for a two-hour US TV special to raise money for victims of the Asian tsunami. 7. Premiership footballer and record company boss Kevin Campbell has gained a court injunction stopping R&B singer Mark Morrison from releasing an album. 8. Actor Stephen Fry is joining the cast of the forthcoming film adaptation of The Hitchhiker's Guide To The Galaxy. 9. A British author has had the film rights to her children's bestseller snapped up for a seven-figure sum, with Ridley Scott set to direct. 10. New York electro-rock group The Bravery have come top of the BBC News-website's Sound of 2005 poll to find the music scene's most promising new act. 11. Novelist Arthur Hailey was known for his bestselling page-turners exploring the inner workings of various industries. 12. Russian drama The Return (Vozvrashchenie) has been named winner of the BBC Four World Cinema Award. 13. Film stars from across the globe are preparing to walk the red carpet at this year's Bafta award ceremony.
← Entertainment	

TABLE 3: First sentences of documents along the directions in linear HLS. Words of Politics/Business/Entertainment topics are highlighted in different colors (same colors in Fig. 8).

the the word space. In particular, [12] uses word embeddings as the vectors and trains a global GMM based on vectors from all documents as a mixture of topics. The means and covariances of the global GMM are then fixed, and the component priors are learned for each document resulting in its topic representation. We follow a similar setting to model each document as one GMM, but we allow each document to have its own Gaussian components instead of fixing the component means and covariances to be the same across documents. The 100-dimension word vectors are trained on Wikipedia corpus using word2vec. We use documents of three topics, i.e. politics, business and entertainment, from the BBC news dataset [48]. There are only around 200 words in each document, thus learning a 100-dim GMM using only one document will lead to ill-conditioned full covariance matrices. Thus, we first learn a global GMM with diagonal covariances using all word vectors from all documents and then use that global GMM as the initialization when estimating each individual document's GMM. Finally we have 300 GMMs of 5 components representing 300 documents in the data. The parameterizations of diagonal covariances for PRIMAL-L/NL are neural networks and we use normal parameterizations for priors and means in linear HLS case and try different combinations of parameterizations for kernel HLS case. The $\boldsymbol{\Theta} = \{\boldsymbol{A}^{(NN)}, \{\{\boldsymbol{m}_{rl}\}_{l=1}^{d_z}, \boldsymbol{b}_r, \boldsymbol{C}_r^{(NN)}\}_{r=1}^{K_m}\}$ performs best in PRIMAL-NL experiment. We compare PRIMAL with embedding models as well as topic models to better position our model in both research fields.

4.6.1 Comparison with embedding models

Fig. 5-3 shows the latent spaces of our methods and baselines, where FINE and PRIMAL-L/NL show relatively good qualitative results. Quantitatively, Table 1 shows that PRIMAL-L/NL achieves the lowest reconstruction loss and also the best LDA classification accuracy. We visualize the manifold by sampling words with high likelihood in each topic GMM and showing their relative change of GMM likelihood along directions in HLS. For example, Fig. 8-1b shows the word likelihood variations from politics and business topics along a line from business to politics region. The y-axis is the log of probability ratio between current and the first word, thus a positive y means a larger likelihood while a negative one means a smaller likelihood. The manifold visualization demonstrates the strength of our models in three ways: 1) the word variations are reasonable, i.e., when the position in HLS changes from topic A to topic B, likelihood of words from topic A all decrease while those from topic B increase; 2) in one topic's region, the relative density values of some words manifest the content of the document, like the high likelihood of music-related words in Fig. 8-2b may indicate that the document is about music entertainment; 3) the kernel HLS (See Appendix D for the visualization of PRIMAL-NL) has more variations along the low-dimensional direction than the linear HLS, which means that the functions learned by PRIMAL-NL are more complex than PRIMAL-L's. We show the actual change in documents in linear HLS in Table 3 by sampling the closest documents when moving along the directions from politics to business and from business to entertainment. The change of sentences reflects the gradual transition between topics.

Fig. 9 shows the sensibility of the HLS variables with respect to the value of k in LDA. PRIMAL-L/NL always performs better than the two baselines, indicating better robustness of HLS variables inferred by PRIMAL.

4.6.2 Comparison with topic models

To further understand the effectiveness of PRIMAL on topic models, we compare the coherence of topics learned by PRIMAL and those learned by two traditional topic models: RecoverKL [51] and Geometric Dirichlet Mean (GDM) [52]. The metric we use is Pointwise Mutual Information (PMI) computed on all Wikipedia documents to obtain reliable coherence statistics following [53]. PMI measures the co-occurrence of any two words from the top-10 topic words in a corpus, i.e, $PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$. There are 55 word pairs in the top-10 topic words, and the PMI for the topic is the average of the 55 word-pair PMIs.

For classical topic models, the PMI evaluation is straightforward since different documents have a common topic

PRIMAL				
Topic 1	*Topic 2*	*Topic3*	*Topic 4*	*Topic 5*
year	film	michael	told	people
one	blair	howard	part	government
first	music	tony	added	years
two	uk	john	before	labour
best	company	david	set	election
time	bbc	bill	back	minister
three	show	charles	says	british
get	plans	martin	former	prime
number	brown	gordan	expected	against
think	market	jamie	including	party
0.6399	0.9237	1.4400	0.9314	1.0558
RecoverKL				
Topic 1	*Topic 2*	*Topic3*	*Topic 4*	*Topic 5*
film	club	film	number	blair
year	year	festival	one	election
one	glazer	year	album	government
music	company	one	year	labour
people	united	director	top	year
goverment	bid	best	best	minister
company	offer	first	first	brown
uk	sales	award	three	people
show	shares	life	film.	prime
blair	board	show	uk	one
0.6072	0.9454	0.7783	0.5449	0.9231

TABLE 4: Top-10 words of 5 topics from PRIMAL and RecoverKL on BBC news. The last rows show the mean PMI scores for the given topic.

subspace and the word probabilities for each topic are often explicit. However, PRIMAL has different topics (GMMs) for different documents and word probabilities are implicit. To obtain a reasonable topic coherence evaluation, we use the mean of each cluster in the latent space of Fig. 5-d3 to generate 3 GMMs as representative documents. As each GMM consists of 5 components corresponding to 5 topics, we evaluate 15 topics in total for PRIMAL. The probability of word w_i under topic t_j is calculated as $p(w_i|t_j) = \frac{p(t_j|w_i)p(w_i)}{\mathcal{Z}}$, where $p(t_j|w_i) = \pi_j \mathcal{N}(\boldsymbol{w}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, $\boldsymbol{w}_i$ is the word embedding of word i, and $\mathcal{Z} = \sum_i p(t_j|w_i)p(w_i)$. The prior $p(w_i)$ is w_i's frequency in the training documents. For RecoverKL and GDM, we set the topic number to 5, corresponding to the 5 topics for each document in our GMM representation.

The mean PMI scores for the three methods are presented in Table 5, and indicate that our learned GMM manifold has better topic coherence than RecoverKL, but worse than GDM. We visualize several topics of PRIMAL and RecoverKL in Table 4. PRIMAL learns more coherent topics, while some topics (*Topic 1*) from RecoverKL are mixes of two different topics (politics and entertainment). PRIMAL also learns a "name" topic (*Topic 3*), which is not captured by the two baseline methods. Finally, we evaluate the topic separation in the latent space. The LDA test accuracy is shown in Table 5, with PRIMAL achieving a higher LDA accuracy than GDM. Thus, PRIMAL is good at topic coherence and latent space interpretation at the same time compared to RecoverKL and GDM. See Appendix D for the latent space visualization.

To compare the scalability and topic identifiability of PRIMAL with GDM and RecoverKL, we run another experiment on a larger topic modeling dataset, 20 Newsgroups [**?**], which comprises 4353 training documents with a vocabulary of 60k words. For the scalability, PRIMAL uses gradient descent so it generally takes longer than GDM and RecoverKL in training (67 minutes for PRIMAL versus 3 and 26 minutes for GDM and RecoverKL). On the other hand,

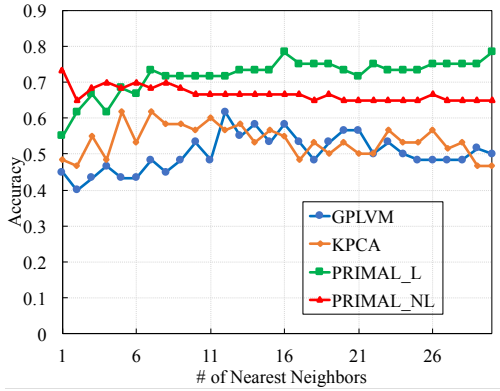


Fig. 9: LDA classification accuracy vs. no. of nearest neighbors.

	RecoverKL [51]	GDM [52]	PRIMAL-L
PMI coherence	0.7598	1.2746	0.9982
LDA accuracy	81.76%	73.33%	78.33%

TABLE 5: Comparisons of PRIMAL with topic models on pointwise mutual information (PMI) and LDA accuracy.

PRIMAL has a space complexity of $O(NDK)$, where D is the dimension of the word vectors, which is typically lower than RecoverKL, which maintains a non-sparse matrix of size V^2, where V is the vocabulary size. Note that adopting stochastic gradient descent will decrease the space complexity of PRIMAL to $O(BDK)$, where B is the batch size. For the identifiability, the topics learned by PRIMAL have an average PMI of 1.1343, while those of GDM and RecoverKL have PMIs of 1.2109 and 0.9317, indicating an advantage of PRIMAL over RecoverKL in learning meaningful topics. See Appendix D for detailed comparisons.

4.6.3 *Training time*

The training time of PRIMAL-L, KPCA, GPLVM and FINE is 958.5s, 0.16s, 288.2s and 1190.5s respectively. Although PRIMAL is a parametric learning method, it uses gradient descent to optimize both the manifold parameters and latent variables, and thus takes more training time than the two non-parametric baselines KPCA and GPLVM. However PRIMAL is still faster than the best non-parametric baseline FINE. Note that PRIMAL is not proposed to be a fast online manifold learning method, but as an interpretable offline data analysis tool. To design a fast and interpretable manifold learning method for GMM distributions is an interesting future work.

5 Conclusion

We propose a parametric manifold learning framework for Gaussian mixture models to obtain a continuous and interpretable low-dimensional latent space. The parametric functions are learned by minimizing the distance between a generative GMM manifold from the latent space and the ground-truth GMM manifold, which is measured by a variational upper bound of KL divergence. We adopt a variational EM optimization algorithm to learn the parameters and HLS variables. We demonstrate the effectiveness of PRIMAL in both synthetic data and several real-world applications. Future work will extend PRIMAL to more complex probabilistic models like hidden Markov models and increase its scalability to larger scale datasets.

ACKNOWLEDGMENTS

This work is supported by a Strategic Research Grant from City University of Hong Kong (Project No. 7005218), and a grant from the Research Grant Council of Hong Kong (GRF No. 17609117).

REFERENCES

[1] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[2] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler, "Hidden Markov models in computational biology: Applications to protein modeling," *Journal of molecular biology*, vol. 235, no. 5, pp. 1501–1531, 1994.

[3] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Intl. J of Comp. Vis.*, vol. 51(2), pp. 91–109, 2003.

[4] G. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[5] D. M. Titterington, *Statistical analysis of finite mixture distributions*. John Wiley & Sons, 1985.

[6] F. Perronnin, C. Dance, G. Csurka, and M. Bressan, "Adapted vocabularies for generic visual categorization," in *Euro. Conf. Computer Vision*, 2006, pp. 464–475.

[7] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, Dec 2013.

[8] P. Perez, J. Sanchez, C. Schmid, M. Douze, H. Jegou, and F. Perronnin, "Aggregating local image descriptors into compact codes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 1704–1716, 09 2012.

[9] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *IEEE Intl. Conf. Computer Vision*, 2005, pp. 1800–1807 Vol. 2.

[10] D. Contractor, P. Singla *et al.*, "Entity-balanced Gaussian pLSA for automated comparison," in *Conf. of North American Chapter of the Assn. for Computational Linguistics: Human Language Technologies*, 2016, pp. 69–79.

[11] R. Das, M. Zaheer, and C. Dyer, "Gaussian LDA for topic models with word embeddings," in *Annual Meeting of the Assn. for Comp. Linguistics (Vol. 1: Long Papers)*, 2015, pp. 795–804.

[12] B. Jiang, Z. Li, H. Chen, and A. G. Cohn, "Latent topic text representation learning on statistical manifolds," *IEEE Trans. Neural Networks and Learning Systems*, no. 99, pp. 1–12, 2018.

[13] Y. Zhao, L. Zhang, and K. Tu, "Gaussian mixture latent vector grammars," in *Annual Meeting of the Assn. for Comp. Linguistics (Vol. 1: Long Papers)*, 2018, pp. 1181–9.

[14] L. Yu, T. Yang, and A. B. Chan, "Density-preserving hierarchical EM algorithm: Simplifying Gaussian mixture models for approximate inference," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 41, no. 6, pp. 1323–1337, 2019.

[15] N. Vasconcelos and A. Lippman, "Learning mixture hierarchies," in *Advances in Neural Information Processing Systems*, 1999, pp. 606–612.

[16] A. B. Chan, E. Coviello, and G. R. Lanckriet, "Clustering dynamic textures with the hierarchical EM algorithm," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 2022–2029.

[17] E. Coviello, G. R. Lanckriet, and A. B. Chan, "The variational hierarchical EM algorithm for clustering hidden Markov models," in *Advances in Neural Information Processing Systems*, 2012, pp. 404–412.

[18] K. M. Carter, R. Raich, W. G. Finn, and A. O. Hero III, "Fine: Fisher information nonparametric embedding," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2093–2098, 2009.

[19] N. Aghaeepour, G. Finak, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, R. H. Scheuermann, F. Consortium, D. Consortium *et al.*, "Critical assessment of automated flow cytometry data analysis techniques," *Nature methods*, vol. 10, no. 3, p. 228, 2013.

[20] C. Y. Chan, A. B. Chan, T. M. Lee, and J. H. Hsiao, "Eye-movement patterns in face recognition are associated with cognitive decline in older adults," *Psychonomic bulletin and review*, pp. 1–8, 2018.

[21] S. T. Roweis, L. K. Saul, and G. E. Hinton, "Global coordination of local linear models," in *Advances in neural information processing systems*, 2002, pp. 889–896.

[22] J.-Y. Kwok and I.-H. Tsang, "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, vol. 15, no. 6, pp. 1517–1525, 2004.

[23] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[24] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[25] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[26] Z. Liu, L. Yu, J. H. Hsiao, and A. B. Chan, "Parametric manifold learning of Gaussian mixture models," in *International Joint Conference on Artificial Intelligence*, 2019.

[27] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," in *USENIX Symp. Operating Systems Design and Implementation*, 2016, pp. 265–283.

[28] J. Goldberger and S. T. Roweis, "Hierarchical clustering of a mixture model," in *Advances in Neural Information Processing Systems*, 2005, pp. 505–512.

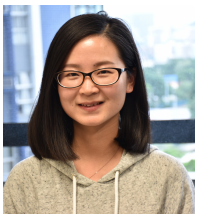
[29] J. V. Davis and I. S. Dhillon, "Differential entropic clustering of multivariate Gaussians," in *Advances in Neural Information Processing Systems*, 2007, pp. 337–344.

[30] A. Mumtaz, E. Coviello, G. R. Lanckriet, and A. B. Chan, "Clustering dynamic textures with the hierarchical EM algorithm for modeling video," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1606–1621, 2012.

[31] E. Coviello, A. B. Chan, and G. R. Lanckriet, "Clustering hidden Markov models with variational HEM," *J.*

Machine Learning Research, vol. 15(1), pp. 697–747, 2014.
[32] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
[33] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications," in *Advances in Neural Information Processing Systems*, 2004, pp. 1385–1392.
[34] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels," *J. Machine Learning Research*, vol. 5, no. Jul, pp. 819–844, 2004.
[35] S.-i. Amari and H. Nagaoka, *Methods of information geometry*. American Mathematical Soc., 2007, vol. 191.
[36] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.
[37] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *J. machine learning research*, vol. 6(Nov), pp. 1783–1816, 2005.
[38] R. M. Altman, "Mixed hidden Markov models: an extension of the hidden Markov model to the longitudinal data setting," *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 201–210, 2007.
[39] D. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Intl. Conf. Learning Representations*, 2014.
[40] E. Richardson and Y. Weiss, "On GANs and GMMs," in *Advances in Neural Information Processing Systems*, 2018, pp. 5847–5858.
[41] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.
[42] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Intl. Conf. Machine Learning*, 1998, pp. 515–521.
[43] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems*, 2006, pp. 1257–1264.
[44] S. Kullback, *Information theory and statistics*. Courier Corporation, 1997.
[45] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–2239, 1998.
[46] P. McCullagh and J. A. Nelder, *Generalized Linear Models*. CRC Press, 1989, vol. 37.
[47] Z. Zhang, G. Dai, D. Wang, and M. I. Jordan, "Bayesian generalized kernel models," in *Intl. Conf. Artificial Intelligence and Statistics*, 2010, pp. 972–979.
[48] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Intl. Conf. Machine learning*, 2006, pp. 377–384.
[49] J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *IEEE Intl. Conf. Acoustics, Speech and Signal Processing*, vol. 4, 2007, pp. IV–317.
[50] T. Chuk, A. B. Chan, and J. H. Hsiao, "Understanding eye movements in face recognition using hidden Markov models," *Journal of vision*, vol. 14(11), 2014.
[51] S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu, "A practical algorithm for topic modeling with provable guarantees," in *International Conference on Machine Learning*, 2013, pp. 280–288.
[52] M. Yurochkin and X. Nguyen, "Geometric Dirichlet means algorithm for topic inference," in *Advances in Neural Information Processing Systems*, 2016, pp. 2505–2513.
[53] D. Newman, S. Karimi, and L. Cavedon, "External evaluation of topic models," in *Australasian Doc. Comp. Symp.*, 2009.
[54] K. Lang, "NewsWeeder: Learning to filter netnews," in *Machine Learning Proceedings 1995*, A. Prieditis and S. Russell, Eds. San Francisco (CA): Morgan Kaufmann, 1995, pp. 331 – 339.

Ziquan Liu is currently a PhD student in the Department of Computer Science, City University of Hong Kong. He received the B.Eng. degree in information engineering and B.S. in mathematics from Beihang Univerisity, Beijing, in 2017. His research interests include low-dimensional representation, deep neural networks and Bayesian inference.



Lei Yu received the BS and MS degree in information and computing science, and computer software and theory from Hunan Normal University, Changsha, China, in 2010 and 2013, and the PhD degree in computer science from City University of Hong Kong. She is currently a postdoc researcher at the University of Hong Kong in the Department of Statistics and Actuarial Science. Her research interests include computer vision, machine learning and optimization.

Janet H. Hsiao is an associate professor at the University of Hong Kong in the Department of Psychology. Before joining HKU, she was a postdoctoral researcher in the Temporal Dynamics of Learning Center (TDLC) at the University of California, San Diego (UC San Diego). She received the Ph.D. degree in Informatics (Cognitive Science) from the University of Edinburgh in 2006. She received the M.Sc. degree in Computing Science from Simon Fraser University in 2002, and the B.Sc. degree in Computer Science from National Taiwan University in 1999. Her research interests include cognitive science, computational modeling, and eye movement analysis.

Antoni B. Chan received the B.S. and M.Eng. degrees in electrical engineering from Cornell University, Ithaca, NY, in 2000 and 2001, and the Ph.D. degree in electrical and computer engineering from the University of California, San Diego (UCSD), San Diego, in 2008. He is currently an Associate Professor in the Department of Computer Science, City University of Hong Kong. His research interests include computer vision, machine learning, pattern recognition, and music analysis.